101665, 745

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 Date of Application:	２００３年　８月２８日
出 願 番 号 Application Number:	特願２００３－３０５５１３
[ST. 10/C]:	［Ｊ Ｐ ２ ０ ０ ３ － ３ ０ ５ ５ １ ３］
出 願 人 Applicant(s):	株式会社リコー

２００３年　９月３０日

特許庁長官
Commissioner,
Japan Patent Office　今 井 康 夫

出証番号　出証特２００３－３０８００５６

【書類名】　特許願
【整理番号】　0306366
【提出日】　平成15年 8月28日
【あて先】　特許庁長官　　殿
【国際特許分類】　G03G 21/00　396
【発明者】
　【住所又は居所】　東京都大田区中馬込１丁目３番６号　株式会社リコー内
　【氏名】　松島　弘幸
【特許出願人】
　【識別番号】　000006747
　【住所又は居所】　東京都大田区中馬込１丁目３番６号
　【氏名又は名称】　株式会社リコー
　【代表者】　桜井　正光
【代理人】
　【識別番号】　100080931
　【住所又は居所】　東京都豊島区東池袋１丁目２０番２号　池袋ホワイトハウスビル８１８号
　【弁理士】
　【氏名又は名称】　大澤　敬
【先の出願に基づく優先権主張】
　【出願番号】　特願2002-276451
　【出願日】　平成14年 9月24日
【先の出願に基づく優先権主張】
　【出願番号】　特願2002-272978
　【出願日】　平成14年 9月19日
【手数料の表示】
　【予納台帳番号】　014498
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　特許請求の範囲　1
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　9809113

【書類名】特許請求の範囲
【請求項１】
第１の通信装置が、第２の通信装置に送信すべき動作要求と前記第２の通信装置から受信した動作要求に対する動作応答とを一括して前記第２の通信装置に送信し、
前記第２の通信装置が、前記第１の通信装置に送信すべき動作要求と前記第１の通信装置から受信した動作要求に対する動作応答とを一括して前記第１の通信装置に送信することを特徴とする通信方法。
【請求項２】
請求項１記載の通信方法であって、
前記動作要求は関数呼び出しであり、
前記動作応答はその関数呼び出しによって呼び出された関数の実行結果であることを特徴とする通信方法。
【請求項３】
請求項１又は２記載の通信方法であって、
前記動作要求及び前記動作応答の送信は、常に前記第２の通信装置から通信要求を発して行い、前記第１の通信装置からの前記動作要求及び前記動作応答の送信は、前記第２の通信装置からの通信要求に対する通信応答として行うことを特徴とする通信方法。
【請求項４】
請求項３記載の通信方法であって、
前記第２の通信装置が前記第１の通信装置に対して定期的に通信要求を行うことを特徴とする通信方法。
【請求項５】
第１の通信装置が、第２の通信装置に送信すべきＳＯＡＰリクエストと前記第２の通信装置から受信したＳＯＡＰリクエストに対するＳＯＡＰレスポンスとを１つのメッセージに記載して前記第２の通信装置に送信し、
前記第２の通信装置が、前記第１の通信装置に送信すべきＳＯＡＰリクエストと前記第１の通信装置から受信したＳＯＡＰリクエストに対するＳＯＡＰレスポンスとを１つのメッセージに記載して前記第１の通信装置に送信することを特徴とする通信方法。
【請求項６】
請求項５記載の通信方法であって、
前記ＳＯＡＰリクエストには関数呼び出しを記載し、
前記ＳＯＡＰレスポンスはその関数呼び出しによって呼び出された関数の実行結果を記載することを特徴とする通信方法。
【請求項７】
請求項５又は６記載の通信方法であって、
前記第２の通信装置は、前記第１の通信装置に送信すべきＳＯＡＰリクエスト及びＳＯＡＰレスポンスを、常にＨＴＴＰリクエストに記載して送信し、
前記第１の通信装置は、前記第２の通信装置に送信すべきＳＯＡＰリクエスト及びＳＯＡＰレスポンスを、常に前記ＨＴＴＰリクエストに対するＨＴＴＰレスポンスに記載して送信することを特徴とする通信方法。
【請求項８】
請求項７記載の通信方法であって、
前記第２の通信装置が前記第１の通信装置に対して定期的にＨＴＴＰリクエストを送信することを特徴とする通信方法。

【書類名】明細書
【発明の名称】通信方法
【技術分野】
【０００１】
この発明は、複数の通信装置の間で、動作要求及び通信相手から受信した動作要求に対する動作応答の送受信を行う通信方法に関する。
【背景技術】
【０００２】
従来から、通信装置をネットワークを介して接続した通信システムにおいて、通信装置同士で互いにメッセージを交換させることにより、通信相手の装置に対して通知や要求を行わせることが行われている。そして、このようなシステムにおいて、ある装置から別の装置に動作要求としてコマンドを送信して動作を実行させ、送信相手から動作の実行結果を動作応答として返信させることも行われている。
【０００３】
このような技術は、例えば特許文献１に開示されており、この文献には、リモートプロセッサがローカルプロセッサに対して実行されるべきコマンドを指示するメッセージを送信し、そのコマンドに対する応答を受信することが記載されている。
また、この文献には、ローカルプロセッサがファイアウォールの内側に配置されている場合において、ローカルプロセッサからファイアウォールの外側のリモートプロセッサに通信要求を送信し、リモートプロセッサがこの通信要求に対する応答としてローカルプロセッサに対してコマンドを送信するようにすることにより、ファイアウォールの外側から内側に向けてコマンドを送信できるようにする技術も開示されている。
【特許文献１】特開２００１－２７３２１１号公報
【０００４】
また、このような動作要求に関する技術は、通信装置に接続された装置の動作を遠隔制御するシステムにも適用することができる。特許文献２には、ブラインド及び照明を操作する機能を有する遠隔被操作装置に、ユーザからの操作を受け付ける機能を有する遠隔操作装置からコマンドを送信してブラインド及び照明を操作させる遠隔操作システムにこのような技術を適用した例が記載されている。ただし、この文献には、コマンドに対する応答を送信する点は示されていない。
【特許文献２】特開２００２－１３５８５８号公報
【発明の開示】
【発明が解決しようとする課題】
【０００５】
ところで、複数の通信装置間でメッセージを交換する場合において、コマンドを送信する通信装置は、１つとは限らない。複数の通信装置が互いに相手に対してコマンドを送信するようにすることも可能であり、この場合には、コマンドを受け付けた通信装置に、それぞれコマンドの送信元に対して実行結果を返させるようにすることが求められている。そして、このような動作を行う場合、ある通信装置から通信相手の装置に送信する情報としては、通信相手の装置に対するコマンドと、通信相手の装置から受信したコマンドについての実行結果とが考えられる。
【０００６】
従来は、これらのコマンドと実行結果とは別々に送信するようにしていた。しかし、このような方式では、コマンドの送信時と受信したコマンドに対する実行結果の送信時とに、それぞれ別々に通信のコネクションを確立する必要がある。従って、通信のオーバーヘッドが大きくなり、効率性の点で問題があった。
現状では、ネットワークを介した通信をダイヤルアップ接続で行う環境もまだ多く残っており、このような環境においては上記の点が特に問題となる。このような環境では、コネクションの確立に数十秒単位の時間を要することもあり、またコネクションを確立する毎に料金を課金されるので、コネクションを確立する回数が増加するとコストアップにつ

ながるためである。
　この発明は、このような問題を解決し、複数の通信装置が互いに動作要求及び受信した動作要求に対する動作応答を送受信する場合において、通信の効率を上げることを目的とする。
【課題を解決するための手段】
　【０００７】
　上記の目的を達成するため、この発明の通信方法は、第１の通信装置と第２の通信装置とが通信する場合において、第１の通信装置が、第２の通信装置に送信すべき動作要求と上記第２の通信装置から受信した動作要求に対する動作応答とを一括して上記第２の通信装置に送信し、上記第２の通信装置が、上記第１の通信装置に送信すべき動作要求と上記第１の通信装置から受信した動作要求に対する動作応答とを一括して上記第１の通信装置に送信するようにしたものである。
　【０００８】
　このような通信方法において、上記動作要求を関数呼び出しとし、上記動作応答を、その関数呼び出しによって呼び出された関数の実行結果とするとよい。
　さらに、上記動作要求及び上記動作応答の送信を、常に上記第２の通信装置から通信要求を発して行い、上記第１の通信装置からの上記動作要求及び上記動作応答の送信は、上記第２の通信装置からの通信要求に対する通信応答として行うようにするとよい。
　さらにまた、上記第２の通信装置が上記第１の通信装置に対して定期的に通信要求を行うようにするとよい。
　【０００９】
　また、この発明は、第１の通信装置が、第２の通信装置に送信すべきＳＯＡＰリクエストと上記第２の通信装置から受信したＳＯＡＰリクエストに対するＳＯＡＰレスポンスとを１つのメッセージに記載して上記第２の通信装置に送信し、上記第２の通信装置が、上記第１の通信装置に送信すべきＳＯＡＰリクエストと上記第１の通信装置から受信したＳＯＡＰリクエストに対するＳＯＡＰレスポンスとを１つのメッセージに記載して上記第１の通信装置に送信する通信方法も提供する。
　【００１０】
　このような通信方法において、上記ＳＯＡＰリクエストには関数呼び出しを記載し、上記ＳＯＡＰレスポンスはその関数呼び出しによって呼び出された関数の実行結果を記載するようにするとよい。
　さらに、上記第２の通信装置が、上記第１の通信装置に送信すべきＳＯＡＰリクエスト及びＳＯＡＰレスポンスを、常にＨＴＴＰリクエストに記載して送信し、上記第１の通信装置が、上記第２の通信装置に送信すべきＳＯＡＰリクエスト及びＳＯＡＰレスポンスを、常に上記ＨＴＴＰリクエストに対するＨＴＴＰレスポンスに記載して送信するようにするとよい。
　さらにまた、上記第２の通信装置が上記第１の通信装置に対して定期的にＨＴＴＰリクエストを送信するようにするとよい。
【発明の効果】
　【００１１】
　以上のようなこの発明の通信方法によれば、通信相手に対する動作要求と、通信相手から受信した動作要求に対する動作応答とを、一度コネクションを確立するのみで送信することができるので、オーバーヘッドを低減し、通信の効率を上げることができる。
【発明を実施するための最良の形態】
　【００１２】
　以下、この発明を実施するための最良の形態について、図面を参照して説明する。
　まず図１に、この発明を適用する通信システムの構成例を示す。
　この通信システムは、図１に示すように、第１の通信装置１と第２の通信装置２とをネットワーク１０によって接続して構成している。
　そして、第１の通信装置１及び第２の通信装置２は、通信機能を備えたＰＣ等のコンピ

ュータを始め、通信機能及び情報処理機能を備えた各種電子装置として構成することができる。ネットワーク１０としては、インターネットやＬＡＮ（ローカルエリアネットワーク）を始め、有線、無線を問わず、ネットワーク通信が可能な各種通信経路を用いることができる。

【００１３】

また、第１の通信装置１及び第２の通信装置２は、互いの制御管理を行うためのアプリケーションプログラムを実装している。そして、これらの各ノードは、ＲＰＣ（Remote Procedure Call）により、互いの実装するアプリケーションプログラムのメソッドに対する処理の依頼である「動作要求」を送信し、この依頼された処理の結果である「動作応答」を取得することができるようになっている。即ち、第１の通信装置１は、第２の通信装置２への要求（以下、第１の通信装置側要求という）を生成してこれを第２の通信装置２へ引き渡し、この要求に対する応答を取得できる一方で、第２の通信装置２は、第１の通信装置１への要求（以下、第２の通信装置側要求という）を生成してこれを第１の通信装置１へ引き渡し、この要求に対する応答を取得できるようになっている。

なお、ここではメソッドを入力と出力の形式を規定した論理的な関数として定義するものとする。そしてこの場合、動作要求はこの関数を呼び出す関数呼び出し（Procedure Call）となり、動作応答はその関数呼び出しによって呼び出された関数の実行結果となる。

【００１４】

図２に、これらの動作要求と動作応答の関係を示す。

図２（Ａ）は、第１の通信装置１で第２の通信装置２に対する動作要求が発生したケースである。このケースでは、第１の通信装置１が第１の通信装置側動作要求を生成して第２の通信装置２に送信し、これを受け取った第２の通信装置２がその要求に対する動作応答を返すというモデルになる。

【００１５】

図２（Ｂ）は、第２の通信装置２で第１の通信装置１に対する動作要求が発生したケースである。このケースでは、第２の通信装置２が第２の通信装置側要求を生成して第１の通信装置１に送信し、これを受け取った第１の通信装置１がその要求に対する動作応答を返すというモデルになる。

なお、ここではＲＰＣによる引数並びに戻り値の受け渡しのプロトコルとしてＳＯＡＰ（Simple Object Access Protocol）を採用し、上記の動作要求や動作応答は、ここではＳＯＡＰメッセージとして記載するようにしている。

【００１６】

この発明の特徴は、このように複数の通信装置が互いに動作要求及び受信した動作要求に対する動作応答を送受信する場合において、通信相手の装置に送信すべき動作要求とその通信相手の装置から受信した動作要求に対する動作応答とを一括して送信するようにする点である。

そして、実際に動作要求や動作応答を転送するための通信プロトコルとしては、システムの構成に合わせて適当なものを採用することができ、例えばＨＴＴＰ（HyperText Transfer Protocol）やＳＭＴＰ（Simple Mail Transfer Protocol）を採用することができる。

そこで、この発明の通信方法に関して、まず通信プロトコルにＨＴＴＰを採用する場合の実施例について説明し、次にＳＭＴＰを採用する場合の実施例について説明する。

【００１７】

〔ＨＴＴＰを採用する場合の実施例：図３乃至図２１〕

図３に、ＨＴＴＰを採用する場合の実施例を適用する通信システムの構成例を示す。

この通信システムは、図３に示すように、ＨＴＴＰサーバ１２とＨＴＴＰクライアント１１とをインターネット１３によって接続して構成している。ただし、セキュリティを向上させるため、ＨＴＴＰクライアント１１はファイアウォール１４を介してインターネット１３に接続するようにしている。そして、ＨＴＴＰサーバ１２が第１の通信装置に、ＨＴＴＰクライアント１１が第２の通信装置に該当する。

【００１８】
なお、ＨＴＴＰを用いて通信を行う場合、ファイアウォール１４の内側にあるノードに対しては、ファイアウォール１４の外側からは自由にアクセスできず、そのノードからの通信要求（ＨＴＴＰリクエスト）に対する通信応答（ＨＴＴＰレスポンス）という形でしかデータを送信できない。そこで、この通信システムにおいては、ファイアウォール１４の内側にあるノードをＨＴＴＰクライアント１１、外側にあるノードをＨＴＴＰサーバ１２としているのである。従って、これらの各ノードの機能は、これら相互間の通信以外においては、クライアントあるいはサーバに限定する必要はない。

【００１９】
また、ＨＴＴＰサーバ１２及びＨＴＴＰクライアント１１は、図１に示した第１の通信装置１及び第２の通信装置２の場合と同様に、互いの制御管理を行うためのアプリケーションプログラムを実装している。そして、ＲＰＣ（Remote Procedure Call）により、互いの実装するアプリケーションプログラムのメソッドに対する処理の依頼である「動作要求」を送信し、この依頼された処理の結果である「動作応答」を取得することができるようになっている。

【００２０】
図４に、これらの動作要求と動作応答の関係を示す。
図４（Ａ）は、ＨＴＴＰクライアント１１でＨＴＴＰサーバ１２に対する動作要求が発生したケースである。このケースでは、ＨＴＴＰクライアント１１がクライアント側動作要求（以下、「クライアントコマンド」とも呼ぶ）を生成してＨＴＴＰサーバ１２に送信し、これを受け取ったＨＴＴＰサーバ１２がそのコマンドに対する動作応答（以下、「コマンド応答」あるいは単に「応答」とも呼ぶ）を返すというモデルになる。

【００２１】
図４（Ｂ）は、ＨＴＴＰサーバ１２でＨＴＴＰクライアント１１に対する動作要求が発生したケースである。このケースでは、ＨＴＴＰサーバ１２がサーバ側動作要求（以下、サーバコマンドとも呼ぶ）を生成してＨＴＴＰクライアント１１に送信し、これを受け取ったＨＴＴＰクライアント１１がそのコマンドに対する動作応答を返すというモデルになる。
このように、動作要求及び動作応答は、ＲＰＣのレベルではＨＴＴＰクライアント１１とＨＴＴＰサーバ１２との間で対称に取り扱われるものである。しかし、通信のレベルでは対称ではない。

【００２２】
図５にこの通信システムにおける通信シーケンスの例を示す。この通信シーケンスは、この発明の通信方法の実施形態を示すものである。
この図に示すように、この通信システムにおいては、通信は常に、ＨＴＴＰクライアント１１から通信要求としてＨＴＴＰリクエストをＨＴＴＰサーバ１２に送信し、ＨＴＴＰサーバ１２からこの通信要求に対する通信応答としてＨＴＴＰレスポンスをＨＴＴＰクライアント１１に返すという手順で行われる。例えばＨＴＴＰクライアント１１が送信したＨＴＴＰリクエストＸに対してＨＴＴＰサーバ１２がＨＴＴＰレスポンスＸを返し、同じくＨＴＴＰリクエストＹに対してＨＴＴＰレスポンスＹを返すという具合である。

【００２３】
そして、ＨＴＴＰリクエストには、ＨＴＴＰクライアント１１からＨＴＴＰサーバ１２に送信する動作要求であるクライアントコマンドと、ＨＴＴＰサーバ１２からＨＴＴＰクライアント１１に送信されてきたサーバコマンドに対する応答（コマンド応答）とを記載して送信するようにしている。また、ＨＴＴＰレスポンスには、ＨＴＴＰサーバ１２からＨＴＴＰクライアント１１に送信する動作要求であるサーバコマンドと、ＨＴＴＰクライアント１１からＨＴＴＰサーバ１２に送信されてきたクライアントコマンドに対する応答（コマンド応答）とを記載して送信するようにしている。

【００２４】
従って、例えばクライアントコマンドＡは、ＨＴＴＰリクエストＸに記載して転送し、

コマンド応答をそのＨＴＴＰリクエストＸと対応するＨＴＴＰレスポンスＸに記載して転送することができる。しかし、サーバコマンドＣについては、ＨＴＴＰリクエストＸと対応するＨＴＴＰレスポンスＸに記載して転送し、そのコマンド応答は次のＨＴＴＰリクエストであるＨＴＴＰリクエストＹに記載して転送することになる。

【００２５】

また、上記図４（Ａ）のケースでは、クライアントコマンドが生成された後直ちにＨＴＴＰクライアント１１がＨＴＴＰサーバ１２とコネクションを確立し、ＨＴＴＰリクエストにこれを含めて引き渡すことができるが、上記図４（Ｂ）のケースでは、ＨＴＴＰクライアント１１側に設置されたファイアウォール１４がＨＴＴＰサーバ１２からのＨＴＴＰリクエストを遮断するため、ＨＴＴＰサーバ１２側からＨＴＴＰクライアント１１へアクセスしてサーバコマンドを直ちに引き渡すことができない。

【００２６】

なお、クライアントコマンド及びサーバコマンドに対する応答をそれぞれ任意の数ずつ（０でもよい）１つのＨＴＴＰリクエストに記載することができ、サーバコマンド及びクライアントコマンドに対する応答をそれぞれ任意の数ずつ（０でもよい）１つのＨＴＴＰレスポンスに記載することができる。そして、１つのＨＴＴＰリクエスト又はＨＴＴＰレスポンスに記載した内容は、論理的に一括して転送する。

そして、このようにすることにより、必要な情報を転送するために必要なコネクションの回数を減らし、オーバーヘッドを低減して通信の効率化を図っている。

【００２７】

図６にこの通信システムにおける別の通信シーケンスの例を示す。この通信シーケンスも、この発明の通信方法の実施形態を示すものである。

説明のため、図５には極めて単純なシーケンス例を示したが、図６には、各ＨＴＴＰリクエストやＨＴＴＰレスポンスに記載するコマンドやコマンド応答の数が一定でない例を示している。

また、コマンドを受信した場合に、次の送信機会の時点で応答を返す必要もない。例えば、図６に示すクライアントコマンドＢのように、コマンドを記載したＨＴＴＰリクエストＸ′に対応するＨＴＴＰレスポンスＸ′に記載して応答を返さず、後のＨＴＴＰレスポンスＹ′に記載して応答を返すようにしてもよい。

もちろんサーバコマンドについても同様であり、サーバコマンドを記載したＨＴＴＰレスポンスの次のＨＴＴＰリクエストにそのコマンドに対する応答を記載する必要はない。そして、さらに後のＨＴＴＰリクエストに記載して転送すればよい。

【００２８】

ところで、各コマンド及びコマンド応答は、それぞれ独立して生成され、また処理に供されるべきものであるから、上記のような一括転送を行うためには、転送前にこれらのコマンドやコマンド応答を結合し、また転送後に分離する処理が必要となる。次に、ＨＴＴＰクライアント１１及びＨＴＴＰサーバ１２のハードウェア構成と共に、このような処理を行うための機能構成及びその処理の手順について説明する。

【００２９】

図７は、ＨＴＴＰクライアント１１及びＨＴＴＰサーバ１２のハードウェア構成例を示す図である。

この図に示すように、ＨＴＴＰクライアント１１及びＨＴＴＰサーバ１２はそれぞれ、ＣＰＵ３１、ＲＯＭ３２、ＲＡＭ３３、ＳＤ（Secure Digital）カード３４、ネットワークインタフェースカード（ＮＩＣ）３５を備え、これらがシステムバス３６で接続されている。

【００３０】

これら構成要素を更に具体的に説明すると、まずＣＰＵ３１は、ＲＯＭ３２に格納している制御プログラムによってＨＴＴＰクライアント１１又はＨＴＴＰサーバ１２全体を統括的に制御する制御手段である。そして、ＲＯＭ３２は、ＣＰＵ３１が使用する制御プログラムを含む各種固定データを格納している読み出し専用メモリである。

ＲＡＭ３３は、ＣＰＵ３１がデータ処理を行う際のワークメモリ等として使用する一時記憶用メモリである。ＳＤカード３４は、装置の電源がオフになっても記憶内容を保持するようになっている不揮発性メモリである。ＮＩＣ３５は、インターネット１３を始めとするネットワークを介して通信相手と情報の送受信を行うための通信手段である。

【００３１】

図８は、ＨＴＴＰクライアント１１の機能のうち、コマンド及びコマンド応答に関する処理を行うための機能の構成を示す機能ブロック図である。

図８に示す機能のうち、クライアントコマンドプール４１及びサーバコマンドプール４２は、いずれかの書き換え可能な記憶手段に設けられるものである。例えばＳＤカード３４に設けることができるが、ＲＡＭ３３や図示しないＨＤＤ（ハードディスクドライブ）に設けてもよい。クライアントコマンド生成手段４３、サーバコマンド実行結果生成手段４４、送信メッセージ収集手段４５、受信メッセージ分配手段４８の機能は、ＣＰＵ３１によって実現されるものである。また、ＨＴＴＰリクエスト送信手段４６及びＨＴＴＰレスポンス受信手段４７の機能は、ＣＰＵ３１及びＮＩＣ３５によって実現されるものである。

【００３２】

これらの機能についてさらに詳述する。

まず、クライアントコマンドプール４１は、クライアントコマンドと、このコマンドに対する応答と、このコマンドの識別情報とを関連付けて登録するプールである。また、サーバコマンドプール４２は、サーバコマンドと、このコマンドに対する応答と、このコマンドの識別情報とを関連付けて登録するプールである。

クライアントコマンド生成手段４３は、クライアントコマンドを生成し、このコマンドを識別する識別情報（ＩＤ）を割り当て、さらにこのコマンドを管理するための管理情報を付し、これらの情報を関連付けてテーブル形式のクライアントコマンドシートとしてクライアントコマンドプール４１に登録する機能を有する。このうち、クライアントコマンドを生成する部分には、例えばＨＴＴＰクライアント１１に備えるアプリケーションが該当する。また、クライアントコマンド生成手段４３に、ＨＴＴＰサーバ１２に各コマンドを実行させる際の優先順位を、生成したクライアントコマンドに付する機能を設けてもよい。

【００３３】

ここで、図９にクライアントコマンドシートにおけるデータ構造の例を示す。

この図に示すように、クライアントコマンドシートには、「コマンドＩＤ」、「メソッド名」、「入力パラメータ」、「状態」、「クライアントコマンド実行結果の通知先」、および「出力パラメータ」のデータを記憶する領域を設けている。そして、このうち「コマンドＩＤ」、「メソッド名」、および「入力パラメータ」がクライアントコマンド（及びそこに付されたＩＤ）に該当し、「状態」及び「クライアントコマンド実行結果の通知先」が管理情報に該当する。「出力パラメータ」は、ＨＴＴＰサーバ１２から受信するコマンド応答の内容である。

【００３４】

次に、各項目の内容について説明する。

まず、「メソッド名」は、ＨＴＴＰサーバ１２に対するリクエストの内容であり、ＨＴＴＰサーバ１２において呼び出す関数の種類を示す。「入力パラメータ」は、「メソッド名」に付随するデータであり、関数を呼び出す際の引数である。「コマンドＩＤ」は、クライアントコマンドを識別するための識別情報である。「状態」は、クライアントコマンドに関する処理の進行状況を示すデータであり、処理の進行と共に、「未送信」→「応答待ち」→「応答受信済」と遷移していく。

【００３５】

「クライアントコマンド実行結果の通知先」は、そのシートに記載しているクライアントコマンドに対する応答を受信した場合に、その旨を通知して必要な処理を実行させるモジュールを示す参照情報である。参照するモジュールは、クライアントコマンドを生成し

たアプリケーションであることが多いが、必ずしもそうである必要はない。「出力パラメータ」には、コマンド応答を受け取った段階で、その内容を格納する。ＨＴＴＰサーバ１２からのコマンド応答を受け取るまでは空である。

【００３６】

図８の説明に戻ると、サーバコマンド実行結果生成手段４４は、サーバコマンドプール４２からサーバコマンドを読み出して実行するアプリケーションである。そして、サーバコマンドに対する応答を生成し、サーバコマンドのコマンドＩＤと関連付けてサーバコマンドプール４２に登録する機能を有する。なお、ＨＴＴＰサーバ１２から受信したサーバコマンドは、このコマンドを識別するＩＤ及びこのコマンドを管理するための管理情報と関連付けて、テーブル形式のサーバコマンドシートとしてサーバコマンドプール４２に登録しておくようにしている。そして、サーバコマンド実行結果生成手段４４が生成したコマンド応答も、実行したサーバコマンドについてのサーバコマンドシートに登録する。

【００３７】

また、サーバコマンド実行結果生成手段４４に、サーバコマンドプール４２から複数の種類のサーバコマンドを読み出し、各サーバコマンドに対する応答を生成する機能を設けることが考えられる。さらに、サーバコマンドがＨＴＴＰクライアント１１に優先して処理を実行させるための実行優先順位の情報を含む場合には、優先順位の高いものから優先的に読み出して実行する機能を設けることも考えられる。

なお、サーバコマンド実行結果生成手段４４は、アプリケーションそのものではなく、サーバコマンドの実行に必要なアプリケーションを呼び出してコマンドを実行させるモジュールであってもよい。

【００３８】

ここで、図１０にサーバコマンドシートにおけるデータ構造の例を示す。

この図に示すように、サーバコマンドシートには、「コマンドＩＤ」、「メソッド名」、「入力パラメータ」、「状態」、「出力パラメータ」、および「サーバコマンドの通知先」のデータを記憶する領域を設けている。そして、このうち「コマンドＩＤ」、「メソッド名」、および「入力パラメータ」がサーバコマンド（及びそこに付されたＩＤ）に該当し、「状態」及び「サーバコマンドの通知先」が管理情報に該当する。「出力パラメータ」は、サーバコマンドの実行結果であり、ＨＴＴＰクライアント１１が返すコマンド応答の内容となる。

【００３９】

次に、各データの内容について説明する。

まず、「メソッド名」は、ＨＴＴＰクライアント１１に対するリクエストの内容であり、ＨＴＴＰクライアント１１において呼び出す関数の種類を示す。「入力パラメータ」は、「メソッド名」に付随するデータであり、関数を呼び出す際の引数である。「コマンドＩＤ」は、サーバコマンドを識別するための識別情報である。「状態」は、サーバコマンドに関する処理の状態を示すデータであり、処理の進行と共に、「未処理」→「処理完了」→「応答済」、あるいは「未処理」→「処理中」→「処理完了」→「応答済」と遷移していく。「出力パラメータ」には、サーバコマンド実行結果生成手段４４によって生成された応答が格納される。サーバコマンドの実行が終了し、上記の「状態」が「処理完了」となるまでは空である。「サーバコマンドの通知先」は、サーバコマンドの実行を行うモジュールを示す参照情報である。

【００４０】

再び図８の説明に戻ると、送信メッセージ収集手段４５は、サーバコマンド実行結果生成手段４４が生成したコマンド応答とこのコマンド応答に対応するサーバコマンドのコマンドＩＤとを関連付けてサーバコマンドプール４２から読み出すと共に、クライアントコマンド生成手段４３が生成したクライアントコマンドとこのコマンドのコマンドＩＤとを関連付けてクライアントコマンドプール４１から読み出し、これらから送信メッセージを生成する機能を有する。

なお、コマンド応答やクライアントコマンドに実行優先順位が指定されている場合には

、送信メッセージ収集手段４５がそれぞれ実行優先順位の高いものから順に読み出すようにすることが考えられる。

【００４１】

ここで、送信メッセージとは、上記のコマンド応答やコマンドとコマンドＩＤとを、構造化言語であるＸＭＬ（Extensible Markup Language）で、ＳＯＡＰメッセージとして記載したものである。そして、送信メッセージ収集手段４５は、１つのコマンド応答あるいはコマンドにつき、送信メッセージとして１つのＳＯＡＰメッセージを生成する。またこのとき、各コマンドのコマンドＩＤはＳＯＡＰヘッダに記載し、コマンド応答及びクライアントコマンドの内容はＳＯＡＰボディに記載する。ＳＯＡＰによる通信では、ＳＯＡＰヘッダとＳＯＡＰボディとからなるＳＯＡＰエンベロープ（封筒）と呼ばれるメッセージをＸＭＬで記載し、ＨＴＴＰなどのプロトコルで交換することになる。

このようなコマンドやコマンド応答からのＳＯＡＰメッセージの生成は、ＷＳＤＬ（Web Service Description Language）に基づいて生成される所要の変換プログラム（シリアライザ）を実行し、データを直列化することによって行うことができる。

【００４２】

そして、ＨＴＴＰリクエスト送信手段４６は、送信メッセージ収集手段４５が生成した送信メッセージを含むＨＴＴＰリクエストを生成し、ＨＴＴＰサーバ１２に送信する機能を有する。このとき、１つのＨＴＴＰリクエストに送信メッセージをいくつ含めてもよいし、コマンド応答に係る送信メッセージとクライアントコマンドに係る送信メッセージとを任意に混在させることもできる。

そこで、ＨＴＴＰリクエスト送信手段４６は、これらのいずれに係る送信メッセージかに関わり無く、送信メッセージ収集手段４５が生成した全ての送信メッセージを１つのＨＴＴＰリクエストに含めて送信するようにしている。ただし、１つのＨＴＴＰリクエストに含める送信メッセージの数に上限を設けることも考えられる。

【００４３】

ところで、このＨＴＴＰリクエストの送信は、送信メッセージ収集手段４５がクライアントコマンドやコマンド応答等の読み出しを試みた場合には、読み出すデータがなく、結果的に送信すべきＳＯＡＰメッセージを生成しなかった場合にも行うものである。そして、この読み出しの試みは、定期的に行うものとする。例えば、タイマによって６０分毎に読み出すことが考えられる。

このようにするのは、上述のように、ＨＴＴＰサーバ１２からＨＴＴＰクライアント１１に送信したい情報があったとしてもＨＴＴＰクライアント１１から通信を要求しない限り送信できないためである。ＨＴＴＰクライアント１１から何も送信するデータがなかったとしても、定期的にＨＴＴＰサーバ１２に対して通信要求を送信して、ＨＴＴＰサーバ１２からＨＴＴＰクライアント１１に情報を送信する機会を与えることにより、転送の必要な情報が長期間に亘ってＨＴＴＰサーバ１２に滞留してしまうことを防止できる。

【００４４】

なお、送信メッセージ収集手段４５による読み出しと、それに続くＨＴＴＰリクエスト送信手段４６によるＨＴＴＰリクエストの送信とを、定期的なタイミング以外に適宜行ってよいことはもちろんである。例えば、緊急に送信が必要な情報がいずれかのプールに登録された場合に、クライアントコマンド生成手段４３あるいはサーバコマンド実行結果生成手段４４が送信メッセージ収集手段４５にその旨を通知して読み出しを行わせるようにしてもよい。

【００４５】

次に、ＨＴＴＰレスポンス受信手段４７は、ＨＴＴＰサーバ１２からＨＴＴＰレスポンスを受信する機能を有する。そしてここでは、ＨＴＴＰレスポンスには、サーバコマンド及びそのコマンドと関連付けられたコマンドＩＤを含む受信メッセージと、クライアントコマンドに対する応答及びそのコマンドと関連付けられたコマンドＩＤを含む受信メッセージとが、任意に混在して含まれている。

ここで、受信メッセージとは、上記のコマンドや応答とコマンドＩＤとをＳＯＡＰメッ

セージとして記載したものである。

【００４６】

受信メッセージ分配手段４８は、ＨＴＴＰレスポンス受信手段４７が受信したＨＴＴＰレスポンスに含まれるデータを、クライアントコマンドプール４１及びサーバコマンドプール４２に振り分けて登録する機能を有する。

具体的には、サーバコマンド及びそのコマンドと関連付けられたコマンドＩＤとをサーバコマンドプール４２にサーバコマンドシートを設けて登録すると共に、クライアントコマンドに対する応答については、そのコマンドと関連付けられたコマンドＩＤをクライアントコマンドプール４１に記憶しているクライアントコマンドシートのコマンドＩＤと照合して対応するクライアントコマンドを特定し、そのクライアントコマンドについての「出力パラメータ」として登録する。

そしてこのとき、ＨＴＴＰレスポンスを分割してそこに含まれる各受信メッセージを取り出し、そのデータをテーブルへの登録に必要な形式に変換するが、この変換は、ＷＳＤＬに基づいて生成される所要の変換プログラム（デシリアライザ）を実行することによって行うことができる。

【００４７】

次に、このような機能を有するＨＴＴＰクライアント１１がＨＴＴＰサーバ１２に送信するＨＴＴＰリクエストの例を図１１に示す。

このＨＴＴＰリクエストは、図１１に示すように、ボディ部としてＭＩＭＥ（Multipurpose Internet Mail Extension）に従ったマルチパートのメッセージが記載され、この各パートには、それぞれエンティティヘッダが記載されると共に、詳細な図示は省略しているが、ＳＯＡＰエンベロープが埋め込まれている。図１１の例では、ＨＴＴＰリクエストのＨＴＴＰボディには、「MIME_boundary」で区分された各要素が、独立した第１パート、第２パート、第３パート、第４パートを構成しているが、ＨＴＴＰボディに含めることのできるパート数は４つに限られない。０個を含め、いくつでもよい。

ＨＴＴＰリクエストに埋め込まれて引き渡されるＳＯＡＰエンベロープには、クライアントコマンドを記載したものと、サーバコマンドに対する応答を記載したものとがある。

【００４８】

また、このような機能を有するＨＴＴＰクライアント１１がＨＴＴＰサーバ１２から受信するＨＴＴＰレスポンスの例を図１２に示す。

図１２に示すように、このＨＴＴＰレスポンスは、形式としては図１１に示したＨＴＴＰリクエストとＨＴＴＰヘッダ部が異なるのみであり、ボディ部にはＨＴＴＰリクエストの場合と同様に、詳細な図示は省略しているが、ＭＩＭＥに従ったマルチパートのＳＯＡＰエンベロープが記載される。ＳＯＡＰエンベロープの内容については、当然コマンドやコマンド応答の内容に従って異なるものである。

ＨＴＴＰレスポンスに埋め込まれて引き渡されるＳＯＡＰエンベロープには、サーバコマンドを記載したものと、クライアントコマンドに対する応答を記載したものとがある。

【００４９】

次に、これらのＨＴＴＰリクエスト又はＨＴＴＰレスポンスに記載されるパートの具体例を図１３乃至図１６に示す。

図１３に示すのは、クライアントコマンドを記載したパートの例である。

この例においては、まず、エンティティヘッダの部分の「X-SOAP-Type」ヘッダに、このパートに記載されているＳＯＡＰメッセージがＳＯＡＰリクエストであるかＳＯＡＰレスポンスであるかを示す情報を記載している。この例では、値の「Request」により、ＳＯＡＰリクエストであること、すなわちコマンドを記載したＳＯＡＰメッセージであることを示している。

また、「SOAPAction」ヘッダは、ＳＯＡＰリクエストの内容を示すものであり、この例では、「http://www.…」というＵＲＩ（Uniform Resource Identifier）によりリクエストの内容を示している。なお、「SOAPAction」ヘッダは、ＳＯＡＰメッセージがＳＯＡＰレスポンスである場合には付加しないため、メッセージの受信側において、このヘッダの

有無により、ＳＯＡＰメッセージがＳＯＡＰリクエストであるかＳＯＡＰレスポンスであるかを判断することもできる。

【００５０】

そして、「Envelope」タグの属性として、名前空間の宣言を行っている。そしてここでは、ＳＯＡＰで標準として定義されている名前空間の他に、「http://www.foo.com/header」及び「http://www.foo.com/server」のＵＲＩで特定される名前空間の宣言を行っている。従って、「ｎ」の名前空間接頭辞が付されたＸＭＬタグについては「http://www.foo.com/header」のＵＲＩで特定される名前空間に属するタグであることがわかり、「ｎｓ」の名前空間接頭辞が付されたＸＭＬタグについては「http://www.foo.com/server」のＵＲＩで特定される名前空間に属するタグであることがわかる。

【００５１】

またＳＯＡＰヘッダには、「要求ＩＤ」のＸＭＬタグの内容として、このクライアントコマンドのＩＤである「１２３４５」が記載されている。そして、ＳＯＡＰボディには、クライアントコマンドシートの「メソッド名」に記憶されていたメソッドを指定する情報として、「異常通知」タグが記載され、その下位のタグ「エラーＩＤ」や「説明」の要素として、「入力パラメータ」に記憶されていた引数が記載されている。ここでは異常通知の通知内容が記載されている。

【００５２】

図１４に示すのは、クライアントコマンドに対する応答を記載したパートの例である。

この例においては、まず、エンティティヘッダの部分の「X-SOAP-Type」ヘッダの値を「Response」と記載することにより、このパートに記載されているＳＯＡＰメッセージがＳＯＡＰレスポンスであること、すなわちコマンド応答を記載したＳＯＡＰメッセージであることを示している。

また、この例においても、名前空間の宣言は図１３に示した例と同様である。そして、ＳＯＡＰヘッダには、「コマンドＩＤ」のＸＭＬタグの内容として、応答を生成したクライアントコマンドのＩ.Ｄである「１２３４５」が記述されている。ＳＯＡＰボディには、「異常通知」コマンドに対する応答であることを示すための「異常通知Response」タグが設けられ、その下位のタグに、コマンド応答の内容が記載される。ここでは、異常通知を正常に受信した旨の情報が記載されている。そして、この情報がクライアントコマンドシートの「出力パラメータ」の項目に格納される。

【００５３】

図１５に示すのは、サーバコマンドを記載したパートの例である。

この例においても、図１３の場合と同様に、「X-SOAP-Type」ヘッダの値の「Request」により、このパートに記載されているＳＯＡＰエンベロープがＳＯＡＰリクエストであることを示し、「SOAPAction」ヘッダの情報により、ＳＯＡＰリクエストの内容を示している。

【００５４】

また、「Envelope」タグの属性として、名前空間の宣言を行っている点も、図１３の場合と同様である。そしてここでは、ＳＯＡＰで標準として定義されている名前空間の他に、「http://www.foo.com/header」及び「http://www.foo.com/client」のＵＲＩで特定される名前空間の宣言を行っている。

ＳＯＡＰヘッダには、「要求ＩＤ」のＸＭＬタグの内容として、このクライアントコマンドのＩＤである「９８７６５」が記載されている。そして、ＳＯＡＰボディには、サーバコマンドシートの「メソッド名」に記憶されるべきメソッドを指定する情報として、「温度センサ値取得」タグが記載され、その下位のタグ「センサＩＤ」の要素として、「入力パラメータ」に記憶されるべき引数が記載されている。ここではセンサ値を取得するセンサのＩＤが記載されている。

なお、サーバがこのようなコマンドを送信する場合としては、例えば、クライアントからの異常通知を受けて異常の原因を特定しようとする場合等が考えられる。

【００５５】

図１６に示すのは、サーバコマンドに対する応答を記載したパートの例である。
この例においても、図１４の場合と同様に、エンティティヘッダの部分の「X-SOAP-Type」ヘッダの値を「Response」と記載することにより、このパートに記載されているＳＯＡＰメッセージがＳＯＡＰレスポンスであることを示している。
また、この例においても、名前空間の宣言は図１５に示した例と同様である。そして、ＳＯＡＰヘッダには、「コマンドＩＤ」のＸＭＬタグの内容として、応答を生成したサーバコマンドのＩＤである「９８７６５」が記述されている。ＳＯＡＰボディには、「温度センサ値取得」コマンドに対する応答であることを示すための「温度センサ値取得Response」タグが設けられ、その下位のタグに、コマンド応答の内容が記載される。ここでは、値取得を要求されたセンサの示す温度値の情報が記載されている。

【００５６】

次に、以上説明したような構成及び機能を有するＨＴＴＰクライアント１１において実行する処理について、図１７乃至図２１のフローチャートを用いて説明する。これらのフローチャートに示す処理は、ＨＴＴＰクライアント１１のＣＰＵ３１が所要の制御プログラムを実行することによって行うものである。

【００５７】

まず、図１７にメッセージの収集及び分配処理の基本動作のフローチャートを示す。
ＨＴＴＰクライアント１１のＣＰＵ３１は、送信メッセージ収集手段４５がクライアントコマンドやコマンド応答等の読み出しを試みるタイミングになると、図１７のフローチャートに示す処理を開始する。
そして、まずクライアントコマンドの収集処理を行う（Ｓ１１）。この処理は、クライアントコマンドプール４１からＨＴＴＰサーバ１２に送信すべきクライアントコマンドを収集する処理であり、収集したデータからＳＯＡＰエンベロープによるパートを生成する処理を含む。

【００５８】

次に、サーバコマンドに対する応答であるサーバコマンド実行結果の収集処理を行う（Ｓ１２）。この処理は、サーバコマンドプールからＨＴＴＰサーバ１２に送信すべきコマンド応答を収集する処理であり、やはり収集したデータからＳＯＡＰエンベロープによるパートを生成する処理を含む。
その後、ステップＳ１１及びＳ１２の処理で生成したパートを１つにマージして、すべてのパートを含むＨＴＴＰリクエストを生成し（Ｓ１３）、そのＨＴＴＰリクエストをＨＴＴＰサーバ１２に送信する（Ｓ１４）。
ここまでの処理において、ステップＳ１１及びＳ１２ではＣＰＵ３１は送信メッセージ収集手段４５として機能し、ステップＳ１３及びＳ１４ではＨＴＴＰリクエスト送信手段４６として機能する。

【００５９】

次に、ＨＴＴＰリクエストに対する通信応答としてＨＴＴＰサーバ１２からＨＴＴＰレスポンスを受信する（Ｓ１５）。そして、受信したＨＴＴＰレスポンスのＨＴＴＰボディを各パートに分割する（Ｓ１６）。ここで、各パートへの分割は、「MIME_boundary」で区分された要素に分割することであり、またここで全てのパートに関して分割する。
そしてその後、分割して得た全てのパートを順に対象として、ステップＳ１７乃至Ｓ１９の処理を繰り返す。この処理においては、まず対象のパートがサーバコマンドを記載したパートか否か判断する（Ｓ１７）。そして、サーバコマンドであればサーバコマンド登録処理を行う（Ｓ１８）。また、サーバコマンドでないときは、クライアントコマンドに対する応答が記載されたパートであるので、応答通知処理を行う（Ｓ１９）。

【００６０】

ステップＳ１８又はＳ１９の後は、ステップＳ１７に戻り、次のパートを対象として処理を繰り返す。そして、全てのパートについてこれらの処理を行った時点で、図１７のフローチャートに示す処理を終了する。
ここまでの処理において、ステップＳ１５及びＳ１６ではＣＰＵ３１はＨＴＴＰレスポ

ンス受信手段４７として機能し、ステップＳ１７乃至Ｓ１９では受信メッセージ分配手段４８として機能する。

【００６１】

次に、図１７のフローチャートに示した処理について、一部分ずつより詳細に示したフローチャートを用いて説明する。

図１８は、図１７のステップＳ１１乃至Ｓ１４の部分の処理をより詳細に示したフローチャートである。

【００６２】

この処理においては、ＨＴＴＰクライアント１１のＣＰＵ３１はまず、クライアントコマンドプール４１から、「状態」が「未送信」であるクライアントコマンドシートの「メソッド名」と「入力パラメータ」の内容を、送信すべきクライアントコマンドとして収集し、「コマンドＩＤ」の内容もそのコマンドのコマンドＩＤとして収集する（Ｓ２１）。「未送信」という「状態」は、コマンドがクライアントコマンド生成手段４３によって生成された後、まだＨＴＴＰサーバ１２に通知されていないことを示すものであるので、これを基準にＨＴＴＰサーバ１２に送信すべきコマンドを抽出できる。

【００６３】

その後、ステップＳ２１で収集した全てのクライアントコマンドを順次対象として、ステップＳ２２乃至Ｓ２４の処理を繰り返す。これらの処理においては、まず対象のクライアントコマンドとそのコマンドＩＤとを、これらの情報がそれぞれＳＯＡＰボディとＳＯＡＰヘッダとに含まれるＸＭＬ文書に変換し（Ｓ２２）、対象のコマンドに関するパートとなるＳＯＡＰエンベロープを生成する（Ｓ２３）。そして、対象のクライアントコマンドを記載していたクライアントコマンドシートの「状態」を「応答待ち」に変更する（Ｓ２４）。「応答待ち」という「状態」は、コマンドをＨＴＴＰサーバ１２に通知済であることを示すものである。

【００６４】

これらが全て完了した後、ＣＰＵ３１は、サーバコマンドプール４２から、「状態」が「処理完了」であるサーバコマンドシートの「出力パラメータ」の内容を、サーバコマンドに対するコマンド応答のうち送信すべきものとして収集し、「コマンドＩＤ」の内容も、対応するサーバコマンドのコマンドＩＤとして収集する（Ｓ２５）。「処理完了」という「状態」は、サーバコマンドに対応する処理がサーバコマンド実行結果生成手段４４によって生成された後、まだＨＴＴＰサーバ１２に通知されていないことを示すものであるので、これを基準にＨＴＴＰサーバ１２に送信すべきコマンド応答を抽出できる。

【００６５】

その後、ステップＳ２５で収集した全てのコマンド応答を順次対象として、ステップＳ２６乃至Ｓ２８の処理を繰り返す。これらの処理は、まず対象のコマンド応答とその応答と共に収集したコマンドＩＤとを、これらの情報がそれぞれＳＯＡＰボディとＳＯＡＰヘッダとに含まれるＸＭＬ文書に変換し（Ｓ２６）、対象のコマンド応答に関するパートとなるＳＯＡＰエンベロープを生成する（Ｓ２７）処理である。これらの処理は、対象が異なる点以外はステップＳ２２及びＳ２３の処理と同じものである。そして、次に対象のコマンド応答を記載していたサーバコマンドシートの「状態」を「応答済」に変更する（Ｓ２８）。「応答済」という「状態」は、コマンド応答をＨＴＴＰサーバ１２に通知済であることを示すものである。

【００６６】

そして、ここまでの処理が全て完了した後、ＣＰＵ３１は、ステップＳ２３又はＳ２７で生成した各パートをマージし、図１１に示したようなマルチパートのＨＴＴＰリクエストを生成してＨＴＴＰサーバ１２に送信する（Ｓ２９）。

なお、ステップＳ２４又はＳ２８で行った「状態」の変更は、実際にこの送信が終了してから行うようにしてもよい。このようにすることにより、通信エラーが発生しても、送信しようとしていたコマンド及びコマンド応答を再度送信の対象とすることができるので、システムの信頼性が向上する。

以上でＨＴＴＰリクエストの送信に関する処理を終了し、図１７のステップＳ１５以降に相当する処理に進む。

【００６７】

図１９は、図１７のステップＳ１５以下の部分の処理をより詳細に示すフローチャートである。図１８のステップＳ２９の次の処理は、この図ではステップＳ３１に該当する。

この処理においては、ＨＴＴＰクライアント１１のＣＰＵ３１はまず、送信したＨＴＴＰリクエストに対するＨＴＴＰレスポンスの受信を待ち、ＨＴＴＰサーバ１２からこれを受信する（Ｓ３１）。これを受信すると、そのＨＴＴＰボディを解析して各パートに分割する（Ｓ３２）。

そしてその後、分割して得た各パートを順次対象として、ステップＳ３３乃至ステップＳ３９の処理を繰り返す。

【００６８】

この部分の処理においては、まず、対象のパートがサーバコマンドであるか否か判断する（Ｓ３３）。上述したように、ＨＴＴＰレスポンスには、クライアントコマンドに対する応答と、サーバコマンドとが含まれている可能性があるので、対象のパートがこのいずれであるかを判断するのである。そして、この判断は、対象のパートにSOAPActionヘッダが存在するか否か、あるいはX-SOAP-Typeヘッダの内容によって判断することができる。

【００６９】

ステップＳ３３でサーバコマンドでなければ、そのパートはクライアントコマンドに対する応答であるので、そのパートのＸＭＬ文書を解析してクライアントコマンドシートに登録できる形式のデータに変換し（Ｓ３４）、クライアントコマンドプール４１からそのコマンド応答に対応するクライアントコマンドを探索し、そのクライアントコマンドについてのクライアントコマンドシートの「出力パラメータ」の項目にコマンド応答のデータを登録する（Ｓ３５）。なお、コマンド応答には、「コマンドＩＤ」の情報として、クライアントコマンドの送信時に付したものと同じコマンドＩＤが付してあるものとし、クライアントコマンドの探索は、この情報をキーとして行うことができる。

【００７０】

データの登録が終わると、データを登録したクライアントコマンドシートの「状態」を「応答受信済」に変更してその旨を示す（Ｓ３６）。そして、「クライアントコマンド実行結果の通知先」に登録されている通知先に、応答があった旨を通知する（Ｓ３７）。この通知によって、クライアントコマンドを生成したアプリケーション等は、その生成したコマンドに応答があったことを認識し、応答に応じた処理を行うことができる。

【００７１】

例えば、異常通知を発するアプリケーションがＨＴＴＰサーバ１２に異常通知を行う旨のクライアントコマンドを生成した場合、このコマンドがＨＴＴＰサーバ１２に送信されると、ＨＴＴＰサーバ１２はこれを正しく受け取った旨のコマンド応答を返してくる。そして、ＨＴＴＰクライアント１１側では、このコマンド応答を受信すると、ここに含まれるコマンドＩＤを基にどのクライアントコマンドに対する応答であるかを探索し、見つかったクライアントコマンドと対応させてそのコマンド応答を登録する。そして、そのコマンドの実行結果通知先として登録されている、異常通知を発するアプリケーションに、応答があった旨を通知するのである。アプリケーションは、この通知を受けた場合にクライアントコマンドシートを参照すれば、生成したコマンドの実行結果を「出力パラメータ」の項目から取得することができる。

以上のステップＳ３７までの処理が終了すると、次のパートがあればそれを対象としてステップＳ３３からの処理を繰り返す。

【００７２】

一方、ステップＳ３３でサーバコマンドであれば、そのパートのＸＭＬ文書を解析してサーバコマンドシートに登録できる形式のデータに変換し（Ｓ３８）、そのサーバコマンドに対応するサーバコマンドシートを作成して、コマンドＩＤと共にサーバコマンドプールに登録する（Ｓ３９）。ここで、サーバコマンドの内容はサーバコマンドシートの「メ

ソッド名」及び「入力パラメータ」の項目に登録し、パートに記載されていたコマンドＩＤは「コマンドＩＤ」の項目に登録する。また、「サーバコマンドの通知先」の項目には、「メソッド名」に記憶させたメソッドを実行させるアプリケーション等への参照情報を、予め用意してあるメソッドとアプリケーション等との対応関係の情報を参照して登録する。「状態」の初期値は「未処理」であり、「出力パラメータ」の初期値はＮＵＬＬである。

【００７３】

以上のステップＳ３９までの処理が終了すると、次のパートがあればそれを対象としてステップＳ３３からの処理を繰り返す。

全てのパートについてステップＳ３３乃至Ｓ３９の処理が終了すると、図１９のフローチャートに示した処理は終了する。

以上のような処理を行うことにより、ＨＴＴＰクライアント１１が、ＨＴＴＰサーバ１２に送信すべき動作要求とＨＴＴＰサーバ１２から受信した動作要求に対する動作応答とを一括してＨＴＴＰサーバ１２に送信することができる。また、ＨＴＴＰサーバ１２からの動作要求とＨＴＴＰサーバ１２に送信した動作要求に対する動作応答とを一括してＨＴＴＰサーバ１２から受信して処理することができる。

【００７４】

なお、ここでは送信すべき全てのパートを全て生成してからマージして送信を行うようにし、また全てのパートを受信してからこれを各パートに分割して処理を行うように説明したが、このようにする必要はない。

送信については、まず始めにＨＴＴＰヘッダを送信し、以後パートを生成するたびにそのパートを順次送信し、全てのパートの送信が完了した時点でその旨のデータを送信するようにしてもよい。このようにしても、これらの課程で送信されるデータが１つのみのＨＴＴＰヘッダを持つ論理的に連続した１つのＨＴＴＰリクエストであれば、１回のセッションで転送でき、ネゴシエーションの処理は１回で済むので、マージして送信する場合と同様な効果を得ることができる。また、送信すべきデータのバッファに必要なメモリ容量を低減できるので、低コストの通信装置で大きなデータを取り扱うことができる。

また、受信側でも、各パートに関する処理を、各パートを受信するたびに順次行うようにすることができる。このようにした場合に容量を低減できることは、送信側の場合と同様である。

【００７５】

次に、サーバコマンドの実行に関する処理について説明する。

図２０は、この処理の一例を示すフローチャートである。

サーバコマンドの実行に関する処理としては、まず、図２０のフローチャートに示す処理を、図１９に示したステップＳ３９の処理の後に、すなわちサーバコマンドをサーバコマンドプール４２に登録した直後に行うことが考えられる。この処理において、ＨＴＴＰクライアント１１のＣＰＵ３１は、サーバコマンド実行結果生成手段４４として機能する。

【００７６】

そして、この処理においては、まず登録したサーバコマンドについてのサーバコマンドシートの「サーバコマンドの通知先」の情報に基づいてアプリケーション等を呼び出し、「メソッド名」や「入力パラメータ」のデータを渡してサーバコマンドに関する処理を実行させる（Ｓ４１）。なお、サーバコマンドに関する処理は、このフローチャートには示していないが、ＣＰＵ３１が別途実行することになる。

そして、これが完了すると、実行結果をサーバコマンドシートの「出力パラメータ」の項目に登録する（Ｓ４２）と共に、サーバコマンドシートの「状態」を「処理完了」に変更し、処理が完了したことを示して（Ｓ４３）、もとの図１９の処理に戻る。

以上の処理を行うことにより、サーバコマンドを実行し、その結果をコマンド応答としてＨＴＴＰサーバ１２に送信可能な状態にすることができる。

【００７７】

また、サーバコマンドの実行に関する処理としては、図１９に示した処理とは独立に、図２１に示す処理を実行することも考えられる。この処理においても、ＨＴＴＰクライアント１１のＣＰＵ３１は、サーバコマンド実行結果生成手段４４として機能する。

この場合、ＣＰＵ３１は適当なタイミングで図２１のフローチャートに示す処理を開始する。

【００７８】

そして、この処理においては、まずサーバコマンドプールに「状態」が「処理待ち」であるサーバコマンドシートがあるか否か判断し（ＳＸ１）、なければこのようなサーバコマンドシートが追加されるまで待機する。そして、このようなサーバコマンドシートを発見した場合、その１つを処理対象とし、そのサーバコマンドシートの「状態」を「処理中」に変更する（ＳＸ２）。

その後、図２０の場合と同様なステップＳ４１乃至Ｓ４３の処理を行って処理対象のサーバコマンドシートに記載されたサーバコマンドを実行し、これが完了するとステップＳＸ１に戻って処理を繰り返す。

【００７９】

以上の処理は、複数のスレッド（例えば４スレッド）で同時に行うようにしてもよい。処理対象となったサーバコマンドシートの「状態」は、「処理待ち」ではないため、複数のスレッドで同時に処理を行っても、１つのサーバコマンドシートを重複して処理対象としてしまうことはない。

【００８０】

以上のような処理を行うようにすれば、任意のタイミングでサーバコマンドを実行することができるので、実行に時間のかかるコマンドがあった場合でも、以後の処理が滞ることがない。そして、実行の終了したものから順に、その結果をコマンド応答としてＨＴＴＰサーバ１２に送信可能な状態にすることができる。

以上で、ＨＴＴＰクライアント１１において実行するこの発明の通信方法に関連する処理の説明を終了する。

【００８１】

なお、ここでは図示及び説明を省略したが、ＨＴＴＰサーバ１２についても、この発明の通信方法に関連する部分においては、ＨＴＴＰクライアント１１と同様なハードウェア構成及び機能構成を有し、同様な処理を行う。ただし、サーバとクライアントという役割の差があることから、以下の点でＨＴＴＰクライアント１１と異なることになる。

【００８２】

まず、ＨＴＴＰリクエスト送信手段４６とＨＴＴＰレスポンス受信手段４７に代えて、ＨＴＴＰレスポンス送信手段とＨＴＴＰリクエスト受信手段を有する。また、クライアントコマンドを受信してこれに対する応答を送信し、サーバコマンドを送信してこれに対する応答を受信することになる。そして、送信メッセージ収集手段４５がコマンドやコマンド応答等の読み出しを試みるタイミングは、ＨＴＴＰリクエストを受信し、その後ＨＴＴＰリクエストの解析が完了した時点である。ＨＴＴＰサーバ１２側では、ＨＴＴＰリクエストに対する通信応答として、コマンドやコマンド応答をＨＴＴＰクライアント１１に送信するためである。

このような処理をＨＴＴＰサーバ１２側でも行うことにより、この発明の通信方法が実現できる。

【００８３】

そして、以上説明してきたこの発明の通信方法によれば、送信元から通信相手に送信すべき動作要求と、通信相手から受信した動作要求に対する動作応答とを、一括して通信相手に送信することができるので、動作要求の送信と動作応答の送信とについて別々にネゴシエーションを行って通信のコネクションを確立する必要がなく、通信のオーバーヘッドを低減して通信効率を高めることができる。

【００８４】

また、動作要求と動作応答とをそれぞれ直列化したデータに変換し、構造化言語形式で

記載した送信メッセージに変換しているので、フォーマットの異なる動作要求と動作応答とを容易に結合し、論理的に１つの送信内容として送信することができる。また、受信側でも、受信した内容を容易に個々のメッセージに分離し、それが動作要求であるか動作応答であるかに応じて適切な処理を行うことができる。

【００８５】

さらに、通信要求を常に一方の装置から発して通信を行い、その通信相手から通信要求元への動作要求等の送信は、その通信要求に対する応答として行うようにすれば、通信要求を発する側の装置がファイアウォールの内側に設けられているような通信システムであっても、ファイアウォールの存在を意識せずに動作要求及び動作応答の送受信を行うことができる。また、通信要求と通信応答とが対応しているため、通信のレベルでのタイミング管理が容易である。

また、この場合において、上記の一方の装置から通信相手に定期的に通信要求を行うようにすれば、ファイアウォールの外側から内側に向けての情報の送信が長時間に亘って停滞してしまう事態を防止できる。

【００８６】

〔ＳＭＴＰを採用する場合の実施例：図２２乃至図２７〕

次に通信プロトコルにＳＭＴＰを採用する場合の実施例について説明する。この実施例は、上述のＨＴＴＰを採用する場合の実施例と共通点が多いので、共通点については説明を省略するか簡単にし、相違点を中心に説明する。

まず、図２２に、ＳＭＴＰを採用する場合の実施例を適用する通信システムの構成例を示す。

【００８７】

この通信システムは、図２２に示すように、通信装置Ａとメールサーバ Ａ′ とを接続したＬＡＮ＿Ａと、通信装置Ｂとメールサーバ Ｂ′ とを接続したＬＡＮ＿Ｂとを、それぞれファイアウォールＡとファイアウォールＢとを介してインターネット１３に接続して構成している。また、ファイアウォールを介して外部からアクセス可能な位置に、ＬＡＮ＿Ａ側ではメールサーバＡを、ＬＡＮ＿Ｂ側ではメールサーバＢをそれぞれ設けている。そして、通信装置Ａが第１の通信装置に、通信装置Ｂが第２の通信装置に該当する。

【００８８】

ＳＭＴＰを用いて通信を行う場合には、通信装置Ａと通信装置Ｂとの間での情報転送は電子メールによって行う。そして、例えば通信装置Ｂから通信装置Ａに情報を送信する場合、図２２に破線で示したように、通信装置Ｂは通信装置Ａを宛先とする電子メールを送信し、これをまずメールサーバ Ｂ′ に送信する。すると、各メールサーバ Ｂ′ ，Ｂ，Ａを順次介して、通信装置Ａが直接アクセスするメールサーバ Ａ′ まで電子メールが転送される。また、図示は省略したが、通常はメールサーバＢとメールサーバＡとの間に更に別のメールサーバを介して転送を行うことになる。

そして、一点鎖線で示したように通信装置Ａが定期的にメールサーバ Ａ′ にアクセスすることにより、通信装置Ａ宛の電子メールを受け取ることができ、以上で通信装置Ｂから通信装置Ａへの情報転送が完了する。通信装置Ａから通信装置Ｂへの情報転送は、逆の手順で行うことができ、情報転送に関しては、通信装置Ａと通信装置Ｂとは対称である。ただし、メールサーバ Ａ′ 及び Ｂ′ を設けることは必須ではなく、通信装置Ａがメールサーバ Ａと、通信装置Ｂがメールサーバ Ｂと直接通信を行うようにしてもよい。また、メール受信用のメールサーバとメール送信用のメールサーバが異なってもよい。

【００８９】

このような通信システムにおいて、各ＬＡＮには外部からアクセス可能な位置にメールサーバを設けていることから、ファイアウォールを越えて電子メールを転送することができる。

なお、この実施例においては、通信装置Ａと通信装置Ｂとが直接ネゴシエーションをした上で通信を行っているわけではないが、相互に情報転送を行うことが可能である。

【００９０】

また、通信装置Ａ及び通信装置Ｂも、図１に示した第１の通信装置及び第２の通信装置の場合と同様に、互いの制御管理を行うためのアプリケーションプログラムを実装している。そして、ＲＰＣにより、互いの実装するアプリケーションプログラムのメソッドに対する処理の依頼である「動作要求」を送信し、この依頼された処理の結果である「動作応答」を取得することができるようになっている。

【００９１】

図２３に、これらの動作要求と動作応答の関係を示すが、動作要求のレベルでは、通信装置Ａと通信装置Ｂとの関係は、既に説明したＨＴＴＰクライアント１１とＨＴＴＰサーバ１２との関係と同様に、対称なものである。しかし、ＳＭＴＰの場合には、通信のレベルでも通信装置Ａと通信装置Ｂとが対称な関係にあることが、ＨＴＴＰの場合と異なる。

【００９２】

図２４に、この通信システムにおける通信シーケンスの例を示す。この通信シーケンスも、この発明の通信方法の実施形態を示すものである。

上述したように、この通信システムにおいては、通信装置Ａと通信装置Ｂとの間の通信は、電子メールを用いて行う。そして、電子メールには、送信元と宛先は存在し、その宛先から送信元に返信を行うことも可能である。しかし、初めの電子メールと返信とは全く独立したものであり、ＨＴＴＰの場合のような、通信要求と通信応答のような関係はない。

従って、どちらの通信装置から先に通信を行ってもよいし、交互に電子メールを送信しなければならないということもないのであるが、ここでは、通信装置Ａから通信装置Ｂにまず電子メールを送信し、交互に計３通の電子メールを送受信する場合の例を示している。

【００９３】

これらの電子メールには、送信先（宛先）に対する動作要求（コマンド）及び、送信先から受信したコマンドに対する動作応答（コマンド応答、あるいは単に「応答」）を記載して送信するようにしている。これは、通信装置Ａと通信装置Ｂのどちらが送信元であっても同様である。

従って、例えば通信装置Ａ側コマンドａは、電子メールｘに記載して転送し、通信装置Ｂからのコマンド応答をその後の電子メールｙに記載して転送することができる。また、通信装置Ｂ側コマンドｃは、電子メールｙに記載して転送し、通信装置Ａからのコマンド応答をその後の電子メールｚに記載して転送することができる。

【００９４】

なお、動作要求及び動作応答に対する応答をそれぞれ任意の数ずつ（０でもよい）１通の電子メールに記載することができる。そして、１通の電子メールに記載した内容は、当然論理的に一括して転送する。そして、このようにすることにより、必要な情報を転送するための電子メールの数を減らし、オーバーヘッドを低減して通信の効率化を図っている。

【００９５】

次に、通信装置Ａ及び通信装置Ｂにおける、コマンドやコマンド応答を結合し、また分離する処理を行うための機能構成及びその処理の手順について説明する。ハードウェア構成については、ＨＴＴＰを用いる実施例の説明において図７に示したＨＴＴＰクライアント１１の場合と同様なものを使用することができる。

【００９６】

図２５は、通信装置Ａの機能のうち、コマンド及びコマンド応答に関する処理を行うための機能の構成を示す図８と対応する機能ブロック図である。

図２５に示す機能のうち、通信装置Ａ側コマンドプール５１，通信装置Ｂ側コマンドプール５２，通信装置Ａ側コマンド生成手段５３，通信装置Ｂ側コマンド実行結果生成手段５４は、それぞれ図８に示したクライアントコマンドプール４１，サーバコマンドプール４２，クライアントコマンド生成手段４３，サーバコマンド実行結果生成手段４４と対応する機能であり、装置の名称が異なることにより名称を変更したものである。

【００９７】
また、送信メッセージ収集手段４５及び受信メッセージ分配手段４８は、図８に示した同名の手段と対応するものである。ただし、送受信するためのデータの形式が、ＳＭＴＰに対応した形式である点が図８に示した例とは異なる。しかし、コマンドやコマンド応答に対応する、個々のＳＯＡＰメッセージについては、図８の場合と同様なものである。
メール送信手段５６及びメール受信手段５７については、送受信に使用するプロトコルが異なることに伴って、図８に示したＨＴＴＰリクエスト送信手段４６及びＨＴＴＰレスポンス受信手段４７とは異なるものである。
【００９８】
すなわち、メール送信手段５６は、送信メッセージ収集手段４５が生成した送信メッセージを含む、通信装置Ｂ宛の電子メールを生成し、メールサーバＡ′に送信する機能を有する。なお、それ以降の電子メールの転送には、通信装置Ａは関与しない。また、１つの電子メールに送信メッセージをいくつ含めてもよいし、コマンド応答に係る送信メッセージと通信装置Ａ側コマンドに係る送信メッセージとを任意に混在させることもできる点等は、図８に示した例の場合と同様である。
【００９９】
メール受信手段５７は、定期的にメールサーバＡ′にアクセスして通信装置Ａ宛ての新着メールの有無を調べ、新着メールがあった場合にこれを受信する機能を有する。そしてここでは、受信する電子メールには、通信装置Ｂ側コマンド及びそのコマンドと関連付けられたコマンドＩＤを含む受信メッセージと、通信装置Ａ側コマンドに対する応答及びそのコマンドと関連付けられたコマンドＩＤを含む受信メッセージとが、任意に混在して含まれている。
【０１００】
次に、このような機能を有する通信装置Ａが通信装置Ｂ宛てに送信する電子メールの例を図２６に示す。
この電子メールは、図１１に示したＨＴＴＰリクエストの場合と同様に、ボディ部としてＭＩＭＥ（Multipurpose Internet Mail Extension）に従ったマルチパートのメッセージが記載され、この各パートには、それぞれＳＯＡＰエンベロープが埋め込まれている。すなわち、ボディの部分はＨＴＴＰリクエストの場合と同様なものになっている。
【０１０１】
しかし、ヘッダ部分はＨＴＴＰリクエストの場合とは異なり、電子メールの送信元アドレスを示すFrom、宛先アドレスを示すTo、表題を示すSubject等の情報が記載されている。
通信装置Ｂから通信装置Ａに送信する電子メールについては、ヘッダのうちFromの内容とToの内容を入れ替えたものになる。
また、これらの電子メールに記載されるＳＯＡＰエンベロープの内容は、ＨＴＴＰリクエストやＨＴＴＰレスポンスの場合と同様なものである。ただし、各パートのエンティティヘッダの部分は、データのエンコード方式が異なるためＨＴＴＰの場合とは一部が異なる。
【０１０２】
次に、以上説明したような構成及び機能を有する通信装置Ａにおいて実行する処理について、図２７及び図２８のフローチャートを用いて説明する。これらのフローチャートに示す処理は、通信装置ＡのＣＰＵが所要の制御プログラムを実行することによって行うものである。
【０１０３】
まず、図２７にメッセージ送信時の処理の基本動作のフローチャートを示す。
通信装置ＡのＣＰＵは、送信メッセージ収集手段４５が通信装置Ａ側コマンドやコマンド応答等の読み出しを試みるタイミングになると、図２７のフローチャートに示す処理を開始する。
そして、まず通信装置Ａ側コマンドの収集処理を行う（Ｓ５１）。この処理は、通信装

置Ａ側コマンドプール５１から通信装置Ｂに送信すべき通信装置Ａ側コマンドを収集する処理であり、収集したデータからＳＯＡＰエンベロープのパートを生成する処理を含む。

【０１０４】

次に、通信装置Ｂ側コマンドに対する応答である通信装置Ｂ側コマンド実行結果の収集処理を行う（Ｓ５２）。この処理は、通信装置Ｂ側コマンドプール５２から通信装置Ｂに送信すべきコマンド応答を収集する処理であり、やはり収集したデータからＳＯＡＰエンベロープのパートを生成する処理を含む。

その後、ステップＳ５１及びＳ５２の処理で生成したパートを１つにマージして、すべてのパートを含む電子メールを生成し（Ｓ５３）、その電子メールを通信装置Ｂに宛てて送信して（Ｓ５４）、処理を終了する。

【０１０５】

以上の処理において、ステップＳ５１及びＳ５２ではＣＰＵ３１は送信メッセージ収集手段４５として機能し、ステップＳ５３及びＳ５４ではメール送信手段５６として機能する。

これらの処理は、図１７に示したステップＳ１１乃至Ｓ１４の処理と対応するものであるが、ＳＭＴＰの場合には、電子メールの送信と受信の間には、通信要求と通信応答のような関係はないため、処理は一旦ここで終了し、電子メールを受信する場合には、別途図２８に示す処理を行う。

【０１０６】

図２８には、メッセージ受信時の処理の基本動作のフローチャートを示す。

通信装置ＡのＣＰＵは、定期的にメールサーバＡ′にアクセスし、通信装置Ａ宛ての新着の電子メールがあると、図２８のフローチャートに示す処理を開始する。

この処理においては、まず新着の電子メールを受信する（Ｓ６１）。そして、受信した電子メールのボディ（本文）を、各パートに分割する（Ｓ６２）。ここで、各パートへの分割は、「MIME_boundary」で区分された要素に分割することであり、またここで全てのパートに関して分解する。

【０１０７】

そしてその後、全てのパートを順に対象としてステップＳ６３乃至Ｓ６５の処理を繰り返す。この処理においては、まず対象のパートが通信装置Ｂ側コマンドを記載したパートか否か判断する（Ｓ６３）。そして、通信装置Ｂ側コマンドであれば通信装置Ｂ側コマンド登録処理を行う（Ｓ６４）。また、通信装置Ｂ側コマンドでないときは、通信装置Ａ側コマンドに対する応答が記載されたパートであるので、応答通知処理を行う（Ｓ６５）。

【０１０８】

ステップＳ６４又はＳ６５の後は、ステップＳ６３に戻り、次のパートを対象として処理を繰り返す。そして、全てのパートについてこれらの処理を行った時点で、図２８のフローチャートに示す処理を終了する。

以上の処理においては、ステップＳ６１ではＣＰＵ３１はメール受信手段５７として機能し、ステップＳ６２乃至Ｓ６５では受信メッセージ分配手段４８として機能する。

これらの処理は、図１７に示したステップＳ１５乃至Ｓ１９の処理と対応するものである。

【０１０９】

また、通信装置Ｂ側コマンドの実行に関する処理については、図２０及び図２１を用いて説明したサーバコマンドの実行に関する処理と同様なものである。

以上で、通信装置Ａにおいて実行する、この発明の通信方法に関連する処理の説明を終了する。

なお、通信装置Ｂの機能及び通信装置Ｂにおいて実行する処理については、通信装置Ａ側要求と通信装置Ｂ側要求との意味合いが反対になる点以外は、通信装置Ａの場合と全く同じであるので、説明を省略する。

【０１１０】

以上説明してきたように、この発明の通信方法は、ＳＭＴＰのように、情報の送信と返

信とが必ずしも対応しないプロトコルを使用して通信を行う場合にも適用できる。そして、この場合にも、送信元から通信相手に送信すべき動作要求と、通信相手から受信した動作要求に対する動作応答とを、一括して通信相手に送信することができるので、動作要求の送信と動作応答の送信とについて別々の単位で転送を行う必要がなく、通信のオーバーヘッドを低減して通信効率を高めることができる。

また、通信に電子メールを用いれば、ファイアウォールの内側へも容易に情報を転送することができる。

【０１１１】

〔各実施例の変形例：図２９〕

次に、上述した各実施例の変形例について説明する。

まず、上述した各実施例では、説明を簡単にするために２つの通信装置からなる通信システムにこの発明を適用する例について説明したが、この発明は、さらに多くの通信装置からなる通信システムに適用することも当然可能である。

例えば、ＨＴＴＰを採用する場合の実施例は、図２９に示すような通信システムにも適用することができる。

【０１１２】

この通信システムは、通信機能を備えた通信装置である管理装置１０２と、同じく通信機能を備えた通信装置である被管理装置１００とをインターネット１３を含むネットワークを介して接続し、管理装置１０２によって被管理装置１００を管理する遠隔管理システムとして構成している。

ここで、被管理装置１００の具体例としては、プリンタ、スキャナ、複写機、あるいはこれらの機能を兼ね備えたデジタル複合機等の画像処理装置を始めとし、通信機能を備えたネットワーク家電，自動販売機，医療機器，電源装置，空調システム，ガス・水道・電気等の計量システム等や、ネットワークに接続可能なコンピュータ等も含め、通信機能を備える各種電子装置が考えられる。

【０１１３】

また、この遠隔管理システムにおいては、管理装置１０２と被管理装置１００との間の通信を仲介する通信装置である仲介装置１０１を設けており、管理装置１０２と被管理装置１００とはこの仲介装置１０１を介して通信を行う。

そして、仲介装置１０１及び被管理装置１００は、その利用環境に応じて多様な階層構造を成す。例えば、図２９に示す設置環境Ａでは、管理装置１０２とＨＴＴＰによるコネクションを確立できる仲介装置１０１ａが、被管理装置１００ａ及び１００ｂを従える単純な階層構造になっているが、設置環境Ｂでは、４台の被管理装置１００を設置するため、１台の仲介装置１０１を設置しただけでは負荷が大きくなる。

【０１１４】

そのため、管理装置１０２とＨＴＴＰによるコネクションを確立できる仲介装置１０１ｂが、被管理装置１００ｃ及び１００ｄだけでなく、他の仲介装置１０１ｃを従え、この仲介装置１０１ｃが被管理装置１００ｅ及び１００ｆを更に従えるという階層構造を形成している。この場合、被管理装置１００ｅ及び１００ｆを遠隔管理するために管理装置１０２から発せられた情報は、仲介装置１０１ｂとその下位のノードである仲介装置１０１ｃとを経由して、被管理装置１００ｅ又は１００ｆに到達することになる。なお、各設置環境には、セキュリティ面を考慮し、ファイアウォール１４を設置している。

【０１１５】

このような遠隔管理システムにおいても、仲介装置１０１をＨＴＴＰクライアント、管理装置１０２をＨＴＴＰサーバとして取り扱うことにより、この発明の通信方法を適用することができる。

なお、この場合において、管理装置１０２から被管理装置１００にコマンドを送信しようとする場合、仲介装置１０１に対して、「被管理装置１００にコマンドを送信せよ」というコマンドを送信し、コマンドに応じた動作として被管理装置１００に対してコマンドを送信させることが考えられる。また、宛先として被管理装置１００のいずれかを指定し

て仲介装置１０１に対してコマンドを送信し、仲介装置１０１に、自己以外が宛先となっているコマンドを、その宛先に対して転送させることも考えられる。

【０１１６】

また、３つ以上のノード間でコマンドやコマンド応答の送受信を行う場合、コマンドの発信元と宛先とを把握できるように、これらの情報もコマンドやコマンド応答のメッセージに含め、またコマンドシートにも記載して管理するようにするとよい。

また、メールサーバを設ければ、ＳＭＴＰを採用する場合の実施例も、このような通信システムに適用することができる。

【０１１７】

さらに、この発明には、上記以外の変形を適用することも可能である。例えば、クライアントコマンドプール４１及びサーバコマンドプール４２に登録するクライアントコマンドシート及びサーバコマンドシートを、ＸＭＬドキュメントとして記載するようにしてもよい。「入力パラメータ」をデシリアライズする前のＸＭＬドキュメントとして保存したり、「出力パラメータ」をシリアライズした後のＸＭＬドキュメントとして保存したりしてもよい。

また、ＨＴＴＰサーバ１２へ送信するコマンドやコマンド応答の情報量に制限を設けても構わない。ＨＴＴＰサーバ１２から受信するコマンドやコマンド応答の情報量に制限を設けても構わない。特に、受信するコマンドの情報量を制限するようにすると、受信側がメモリ容量の限られた装置である場合にメモリの使用量を抑えることができる。

【０１１８】

また、上述した実施例においては、ＲＰＣを実現する上位プロトコルとしてＳＯＡＰを採用し、アプリケーションは直接プールを操作してＲＰＣを実現しているが、アプリケーションとプールとの間にＣＯＲＢＡ（Common Object Request Broker Architecture）やＪＡＶＡ（登録商標）ＲＭＩ（Remote Method Invocation）とのブリッジ（メッセージ変換機能）を備えることによってアプリケーションの開発効率をさらに向上させてもよい。

すなわち、上述した実施例における、ＨＴＴＰクライアント１１とＨＴＴＰサーバ１２との間等でのコマンド及びこれに対するコマンド応答のやり取りは、ＸＭＬで記述されたＳＯＡＰメッセージにより行うこととしているが、これに限るものでなく、他の形式で記述されていてもよい。

【０１１９】

また、上述した実施例において、ＳＯＡＰ標準のプロトコルだけでなく、ＳＯＡＰとＭＩＭＥマルチパートを組み合わせた独自のプロトコルをもこれに加えて採用することにより、ＨＴＴＰリクエスト、或いはＨＴＴＰレスポンスに含まれるＳＯＡＰエンベロープを全く独立したものとして扱うこととするが、ＳＯＡＰの関連仕様として定義されたＳＯＡＰアタッチメントによって、ＨＴＴＰレスポンスに含まれる第１パートのＳＯＡＰエンベロープに、第２パート以降のＳＯＡＰエンベロープへのリンクを埋め込んでこれらを関連付けて引き渡す構成にしてもよい。ＳＭＴＰを用いる電子メールの場合にも同様である。

【０１２０】

更に、ＳＯＡＰ等の上位プロトコルの下位に位置するデータ通信のプロトコルとして、ここではＨＴＴＰあるいはＳＭＴＰを採用した例について説明したが、この下位プロトコルについても、ＦＴＰ（File Transfer Protocol）等の他のプロトコルを採用してもよい。

さらにまた、この発明を適用する通信システムについても、以上説明したものに限られることはない。

【産業上の利用可能性】

【０１２１】

以上説明してきたように、この発明の通信方法によれば、複数の通信装置が互いに動作要求及び受信した動作要求に対する動作応答を送受信する場合において、通信の効率を上げることができる。従って、この発明を、複数の通信装置が互いに動作要求及び受信した動作要求に対する動作応答を送受信する通信システムに適用することにより、通信の負荷

が小さい通信システムを構成することができる。

【図面の簡単な説明】

【０１２２】

【図１】この発明の通信方法を適用する通信システムの構成例を示す図である。

【図２】図１に示した通信システムにおける動作要求と動作応答の関係を示す図である。

【図３】通信プロトコルとしてＨＴＴＰを採用する場合のこの発明の通信方法の実施例を適用する通信システムの構成例を示す図である。

【図４】図３に示した通信システムにおける動作要求と動作応答の関係を示す図である。

【図５】図３に示した通信システムにおける通信シーケンスの例を示す図である。

【図６】その別の例を示す図である。

【図７】図３に示したＨＴＴＰクライアント及びＨＴＴＰサーバのハードウェア構成例を示す図である。

【図８】図３に示したＨＴＴＰクライアントの機能のうち、コマンド及びコマンド応答に関する処理を行うための機能の構成を示す機能ブロック図である。

【図９】図８に示したクライアントコマンドプールに記憶させるクライアントコマンドシートにおけるデータ構造の例を示す図である。

【図１０】図８に示したサーバコマンドプールに記憶させるサーバコマンドシートにおけるデータ構造の例を示す図である。

【０１２３】

【図１１】図３に示したＨＴＴＰクライアントがＨＴＴＰサーバに送信するＨＴＴＰリクエストの例を示す図である。

【図１２】図３に示したＨＴＴＰクライアントがＨＴＴＰサーバから受信するＨＴＴＰレスポンスの例を示す図である。

【図１３】クライアントコマンドを記載したパートの例を示す図である。

【図１４】クライアントコマンドに対する応答を記載したパートの例を示す図である。

【図１５】サーバコマンドを記載したパートの例を示す図である。

【図１６】サーバコマンドに対する応答を記載したパートの例を示す図である。

【図１７】図３に示したＨＴＴＰクライアントにおける、メッセージの収集及び分配処理の基本動作を示すフローチャートである。

【図１８】図１７のステップＳ１１乃至Ｓ１４の部分の処理をより詳細に示すフローチャートである。

【図１９】図１７のステップＳ１５以下の部分の処理をより詳細に示すフローチャートである。

【図２０】図３に示したＨＴＴＰクライアントにおける、サーバコマンドの実行に関する処理の一例を示すフローチャートである。

【０１２４】

【図２１】その別の例を示すフローチャートである。

【図２２】通信プロトコルとしてＳＭＴＰを採用する場合のこの発明の通信方法の実施例を適用する通信システムの構成例を示す図である。

【図２３】図２２に示した通信システムにおける動作要求と動作応答の関係を示す図である。

【図２４】図２２に示した通信システムにおける通信シーケンスの例を示す図である。

【図２５】図２２に示した通信装置Ａの機能のうち、コマンド及びコマンド応答に関する処理を行うための機能の構成を示す、図８と対応する機能ブロック図である。

【図２６】図２２に示した通信装置Ａが通信装置Ｂ宛てに送信する電子メールの例を示す図である。

【図２７】図２２に示した通信装置Ａにおける、メッセージ送信時の処理の基本動作を示すフローチャートである。

【図２８】同じく、メッセージ受信時の処理の基本動作のフローチャートを示すフローチャートである。

【図２９】この発明の通信方法の実施例を適用する通信システムの別の構成例を示す図である。

【符号の説明】

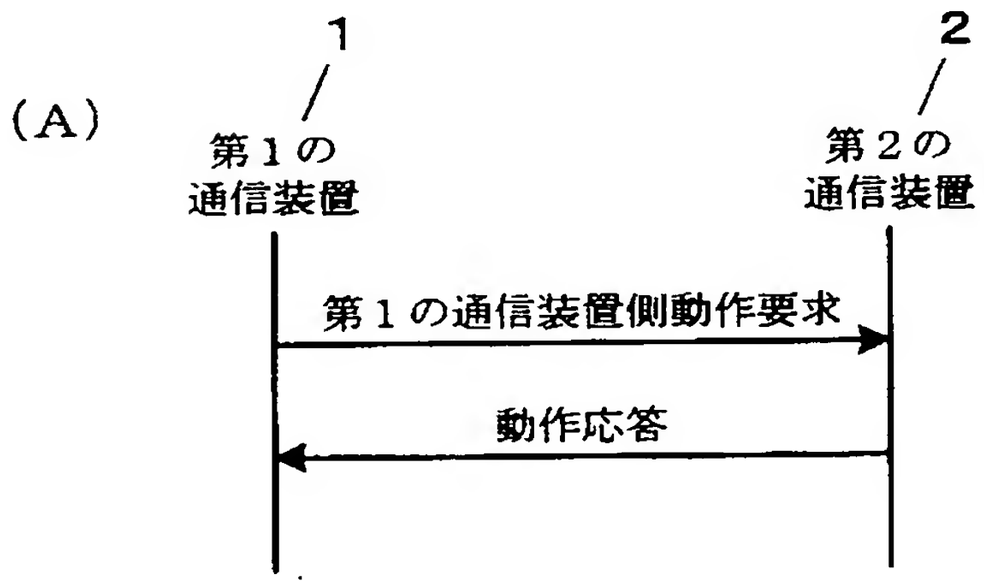
【０１２５】

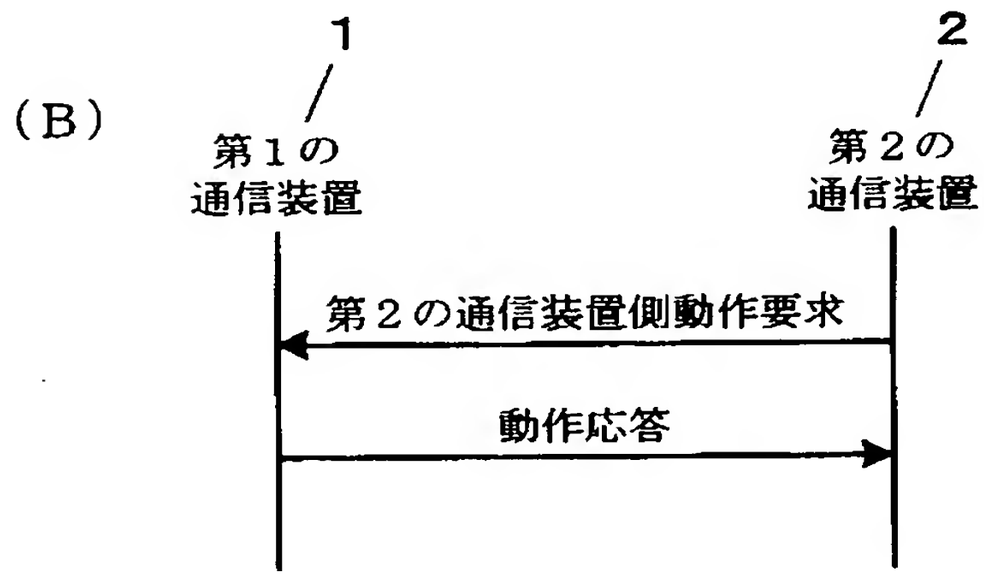
１：第１の通信装置　　２：第２の通信装置
１０：ネットワーク　　１１：ＨＴＴＰクライアント
１２：ＨＴＴＰサーバ　　１３：インターネット
１４：ファイアウォール　　３１：ＣＰＵ
３２：ＲＯＭ　　３３：ＲＡＭ
３４：ＳＤカード　　３５：ＮＩＣ
４１：クライアントコマンドプール
４２：サーバコマンドプール
４３：クライアントコマンド生成手段
４４：サーバコマンド実行結果生成手段
４５：送信メッセージ収集手段
４６：ＨＴＴＰリクエスト送信手段
４７：ＨＴＴＰレスポンス受信手段
４８：受信メッセージ分配手段
５１：通信装置Ａ側コマンドプール
５２：通信装置Ｂ側コマンドプール
５３：通信装置Ａ側コマンド生成手段
５４：通信装置Ｂ側コマンド実行結果生成手段
５６：メール送信手段　　５７：メール受信手段
１００：被管理装置　　１０１：仲介装置
１０２：管理装置

【書類名】図面
【図１】

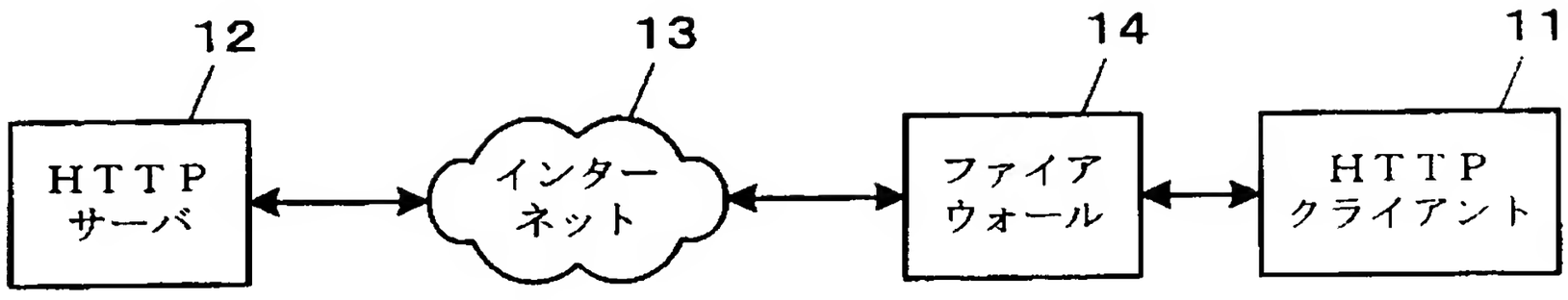

1
10
2
第１の
通信装置
ネット
ワーク
第２の
通信装置

【図２】

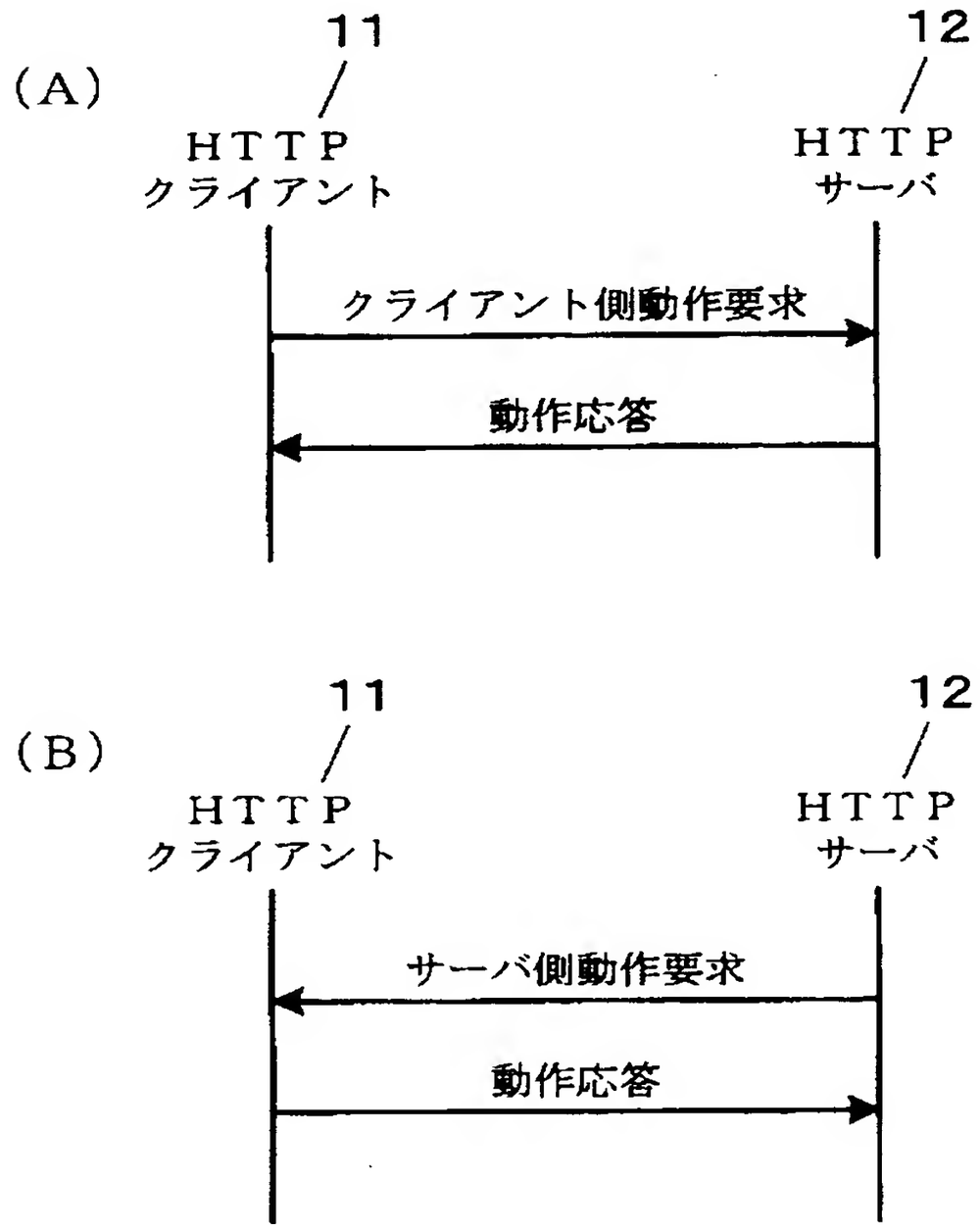

(A)
1
2
第１の
通信装置
第２の
通信装置
第１の通信装置側動作要求
動作応答


(B)
1
2
第１の
通信装置
第２の
通信装置
第２の通信装置側動作要求
動作応答

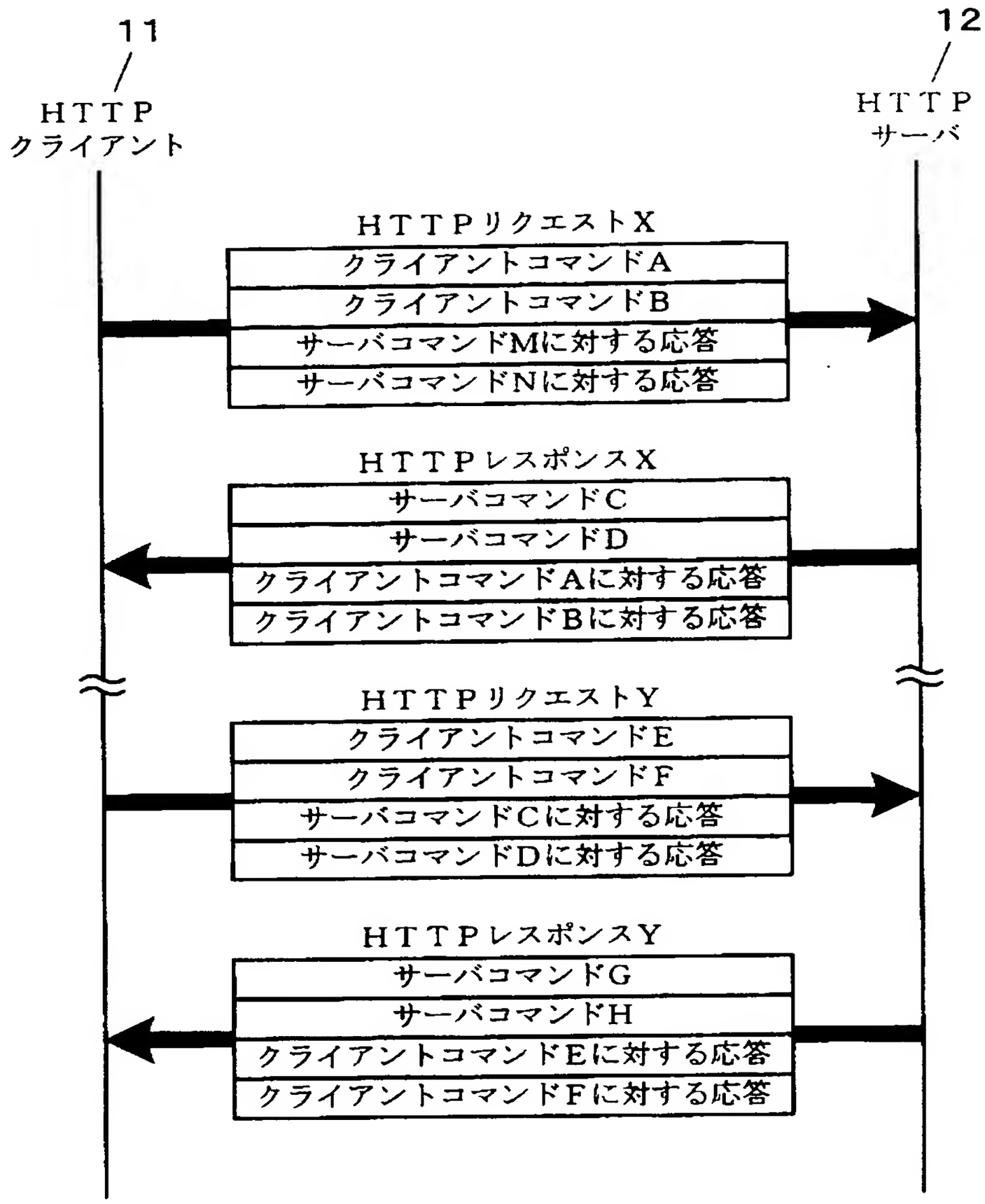
【図３】


12
13
14
11
HTTP
サーバ
インター
ネット
ファイア
ウォール
HTTP
クライアント

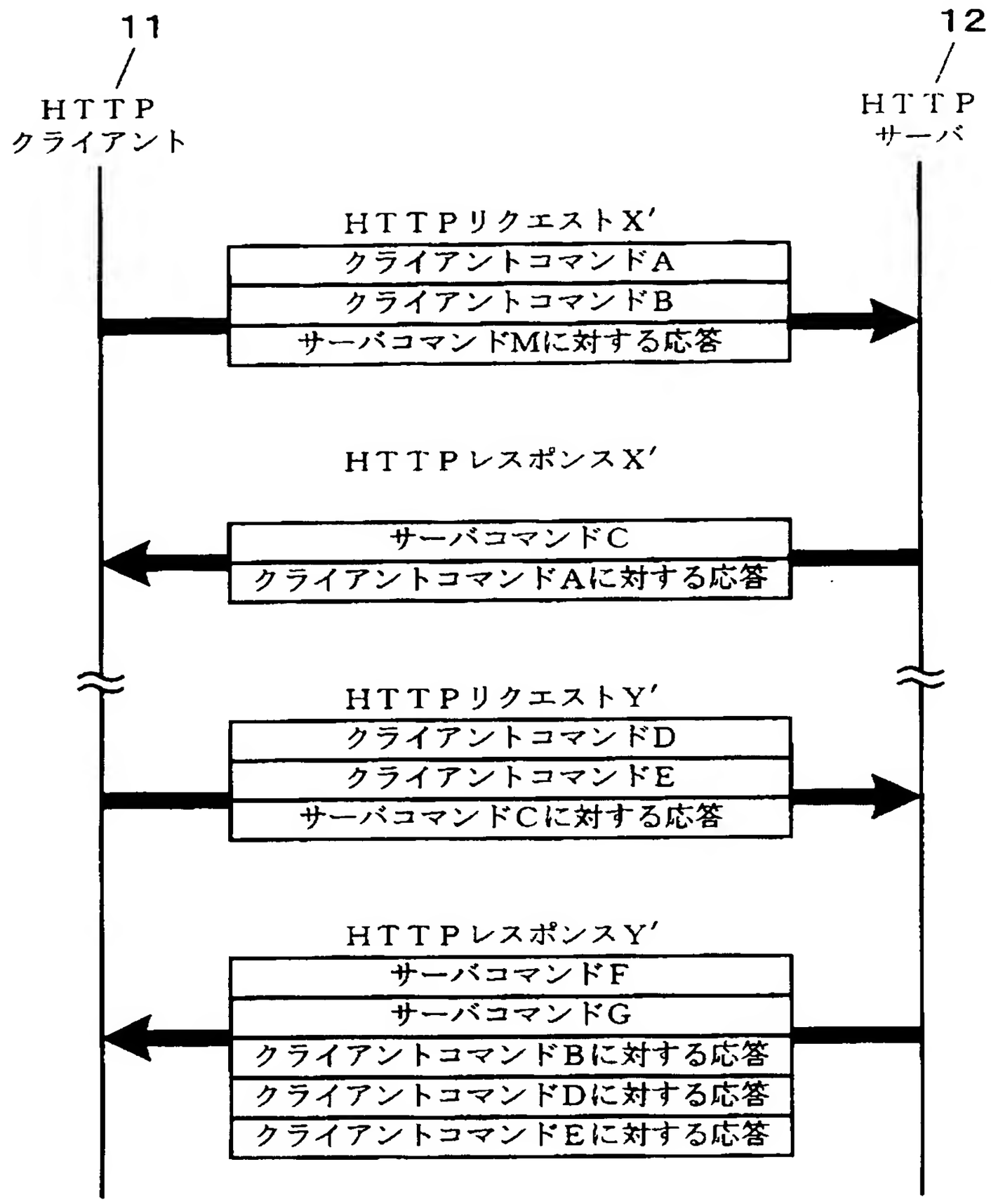
【図４】


(A)
11
12
HTTP
クライアント
HTTP
サーバ
クライアント側動作要求
動作応答
(B)
11
12
HTTP
クライアント
HTTP
サーバ
サーバ側動作要求
動作応答

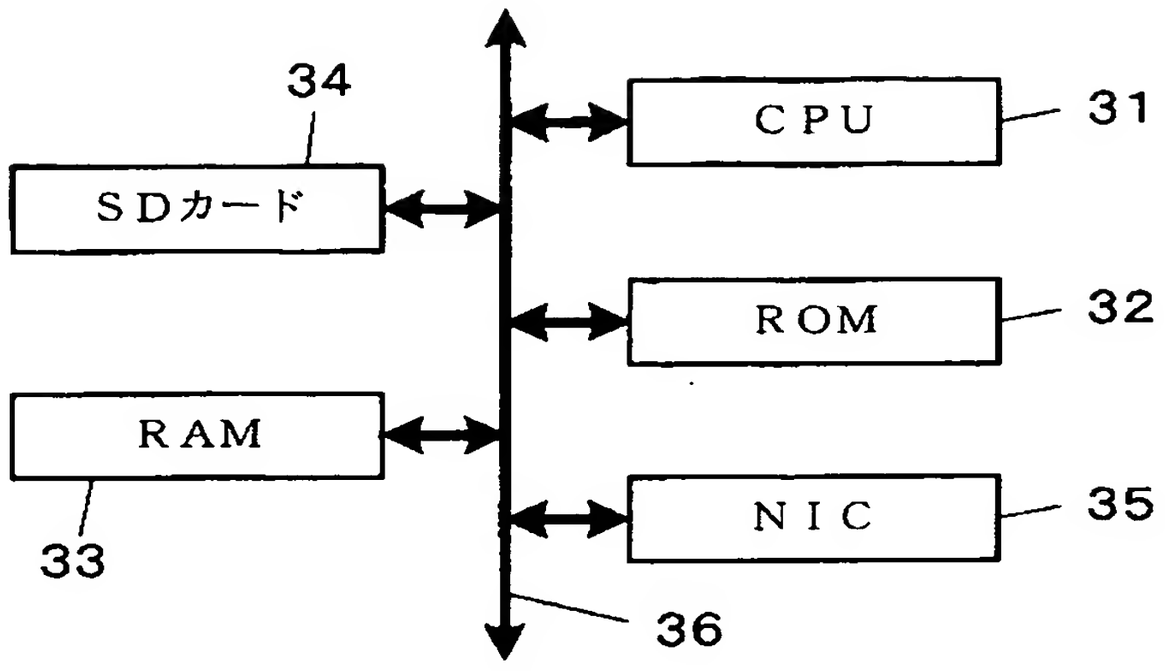
【図５】


11
HTTP
クライアント
12
HTTP
サーバ
HTTPリクエストX
クライアントコマンドA
クライアントコマンドB
サーバコマンドMに対する応答
サーバコマンドNに対する応答
HTTPレスポンスX
サーバコマンドC
サーバコマンドD
クライアントコマンドAに対する応答
クライアントコマンドBに対する応答
HTTPリクエストY
クライアントコマンドE
クライアントコマンドF
サーバコマンドCに対する応答
サーバコマンドDに対する応答
HTTPレスポンスY
サーバコマンドG
サーバコマンドH
クライアントコマンドEに対する応答
クライアントコマンドFに対する応答

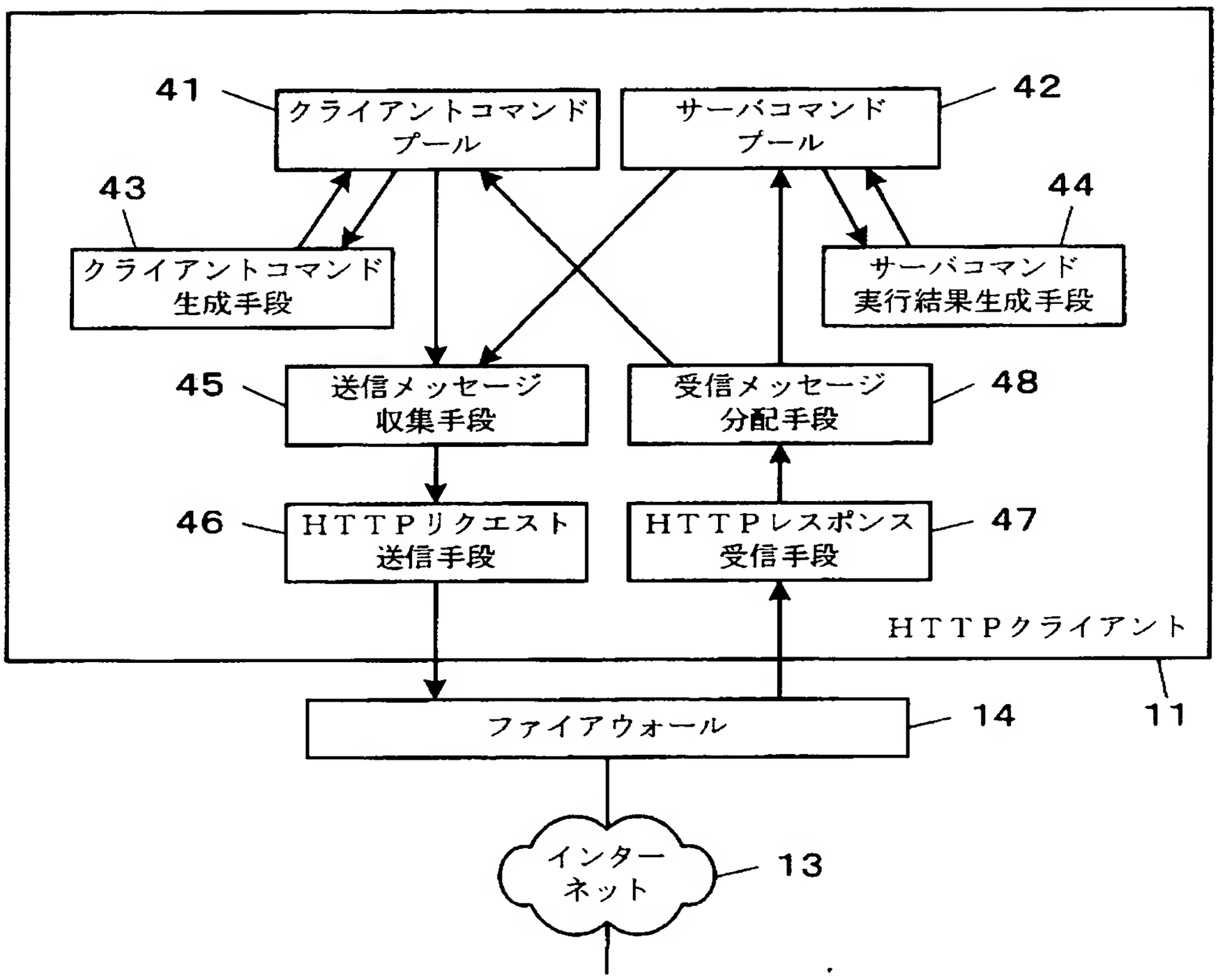
【図６】


11
HTTP
クライアント
12
HTTP
サーバ
HTTPリクエストX′
クライアントコマンドA
クライアントコマンドB
サーバコマンドMに対する応答
HTTPレスポンスX′
サーバコマンドC
クライアントコマンドAに対する応答
HTTPリクエストY′
クライアントコマンドD
クライアントコマンドE
サーバコマンドCに対する応答
HTTPレスポンスY′
サーバコマンドF
サーバコマンドG
クライアントコマンドBに対する応答
クライアントコマンドDに対する応答
クライアントコマンドEに対する応答

【図７】


34
SDカード
CPU
31
ROM
32
RAM
33
NIC
35
36

【図８】


41
クライアントコマンド
プール
42
サーバコマンド
プール
43
クライアントコマンド
生成手段
44
サーバコマンド
実行結果生成手段
45
送信メッセージ
収集手段
48
受信メッセージ
分配手段
46
HTTPリクエスト
送信手段
47
HTTPレスポンス
受信手段
HTTPクライアント
ファイアウォール
14
11
インター
ネット
13

【図９】

クライアントコマンドシート

コマンドＩＤ
メソッド名（異常通知等）
入力パラメータ（異常内容等）
状態（初期値：未処理）
クライアントコマンド実行結果の通知先
出力パラメータ（応答取得まで空）

【図１０】

サーバコマンドシート

コマンドＩＤ
メソッド名（カウンタ取得等）
入力パラメータ
状態（初期値：未処理）
出力パラメータ（処理終了まで空）
サーバコマンドの通知先

【図１１】

ＨＴＴＰリクエスト

```
POST /aaa HTTP/1.1
Content-Type:multipart/mixed;boundary=MIME_boundary
Content-Length:nnnn

--MIME_boundary
Content-Type:text/xml;charset=UTF-8
Content-Transfer-Encoding:8bit
X-SOAP-Type:Request
SOAPAction:"SOAP Action URI"

<s:Envelope>
  <--SOAP Request-->
</s:Envelope>

--MIME_boundary
Content-Type:text/xml;charset=UTF-8
Content-Transfer-Encoding:8bit
X-SOAP-Type:Response

<s:Envelope>
  <--SOAP Response-->
</s:Envelope>

--MIME_boundary
Content-Type:text/xml;charset=UTF-8
Content-Transfer-Encoding:8bit
X-SOAP-Type:Response

<s:Envelope>
  <--SOAP Response-->
</s:Envelope>

--MIME_boundary
Content-Type:text/xml;charset=UTF-8
Content-Transfer-Encoding:8bit
X-SOAP-Type:Response

<s:Envelope>
  <--SOAP Response-->
</s:Envelope>

--MIME_boundary--
```

第１パート

第２パート

第３パート

第４パート

【図１２】

ＨＴＴＰレスポンス

```
HTTP/1.1 200 OK
Content-Type:multipart/mixed;boundary=MIME_boundary
Content-Length:nnnn

--MIME_boundary
Content-Type:text/xml;charset=UTF-8
Content-Transfer-Encoding:8bit
X-SOAP-Type:Request
SOAPAction:"SOAP Action URI"

<s:Envelope>
  <--SOAP Request-->
</s:Envelope>

--MIME_boundary
Content-Type:text/xml;charset=UTF-8
Content-Transfer-Encoding:8bit
X-SOAP-Type:Response

<s:Envelope>
  <--SOAP Response-->
</s:Envelope>

--MIME_boundary
Content-Type:text/xml;charset=UTF-8
Content-Transfer-Encoding:8bit
X-SOAP-Type:Response

<s:Envelope>
  <--SOAP Response-->
</s:Envelope>

--MIME_boundary
Content-Type:text/xml;charset=UTF-8
Content-Transfer-Encoding:8bit
X-SOAP-Type:Response

<s:Envelope>
  <--SOAP Response-->
</s:Envelope>

--MIME_boundary--
```

第１パート

第２パート

第３パート

第４パート

【図１３】

クライアントコマンドのパート例

```
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
X-SOAP-Type: Request
SOAPAction: "http://www.foo.com/server/errorNotification"

<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:se="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:n="http://www.foo.com/header"
  xmlns:ns:"http://www.foo.com/server"
  s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">

  <s:Header>
    <n:要求ＩＤ>12345</n:要求ＩＤ>
  </s:Header>
  </s:Body>
    <ns:異常通知>
      <エラーＩＤ>1111</エラーＩＤ>
      <説明>ハードディスクドライブ異常</説明>
    </ns:異常通知>
  </s:Body>
</s:Envelope>
```

【図１４】

クライアントコマンドに対する応答のパート例

```
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
X-SOAP-Type: Response

<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:se="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:n="http://www.foo.com/header"
  xmlns:ns:"http://www.foo.com/server"
  s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">

  <s:Header>
    <n:要求ＩＤ>12345</n:要求ＩＤ>
  </s:Header>
  </s:Body>
    <ns:異常通知Response>
      <受信結果>OK</受信結果>
    </ns:異常通知Response>
  </s:Body>
</s:Envelope>
```

【図１５】

サーバコマンドのパート例

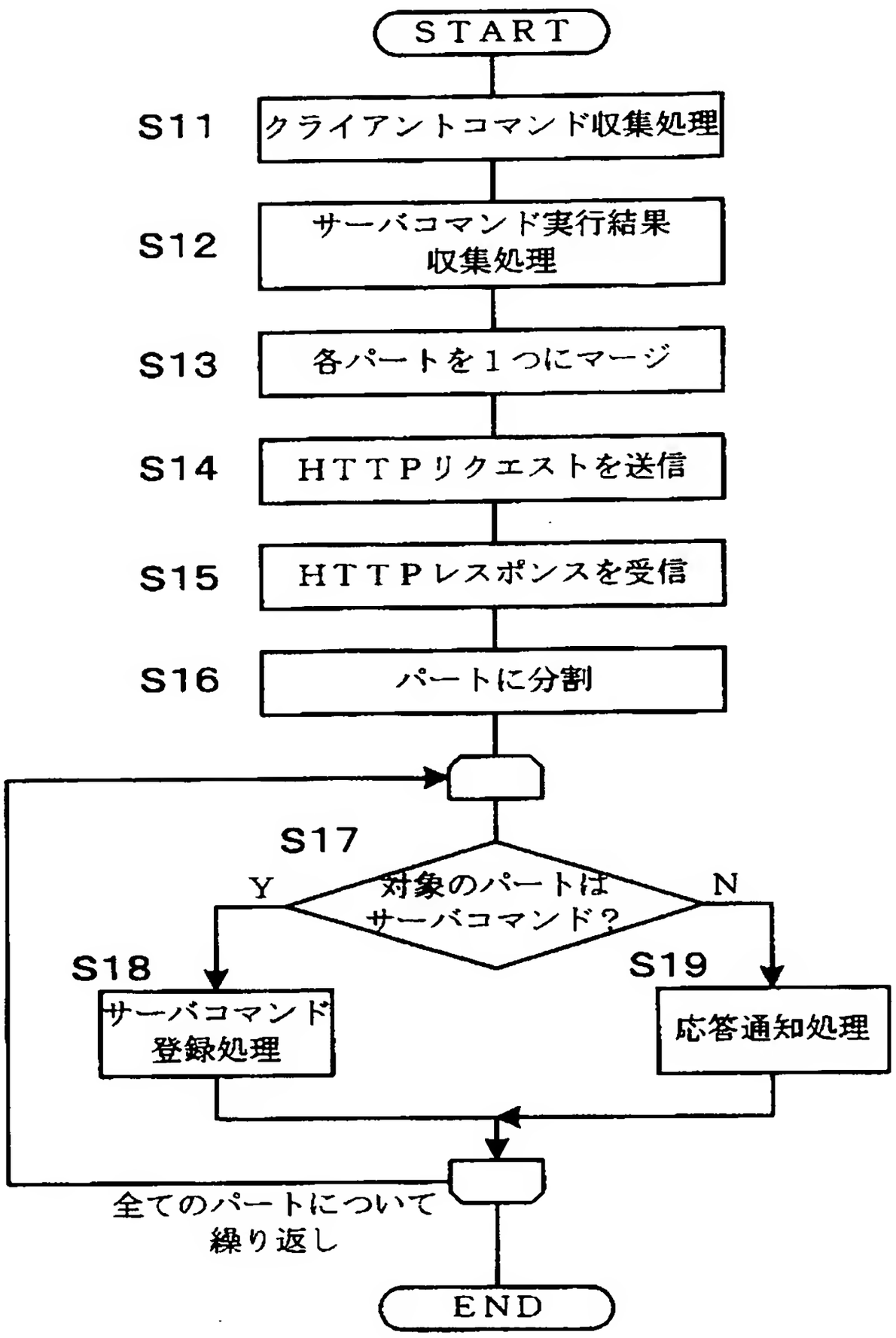
```
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
X-SOAP-Type: Request
SOAPAction: "http://www.foo.com/client/getTemperature"

<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:se="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:n="http://www.foo.com/header"
  xmlns:nc:"http://www.foo.com/client"
  s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">

  <s:Header>
    <n:要求ＩＤ>98765</n:要求ＩＤ>
  </s:Header>
  </s:Body>
    <nc:温度センサ値取得>
      <センサＩＤ>3</センサＩＤ>
    </nc:温度センサ値取得>
  </s:Body>
</s:Envelope>
```

【図１６】

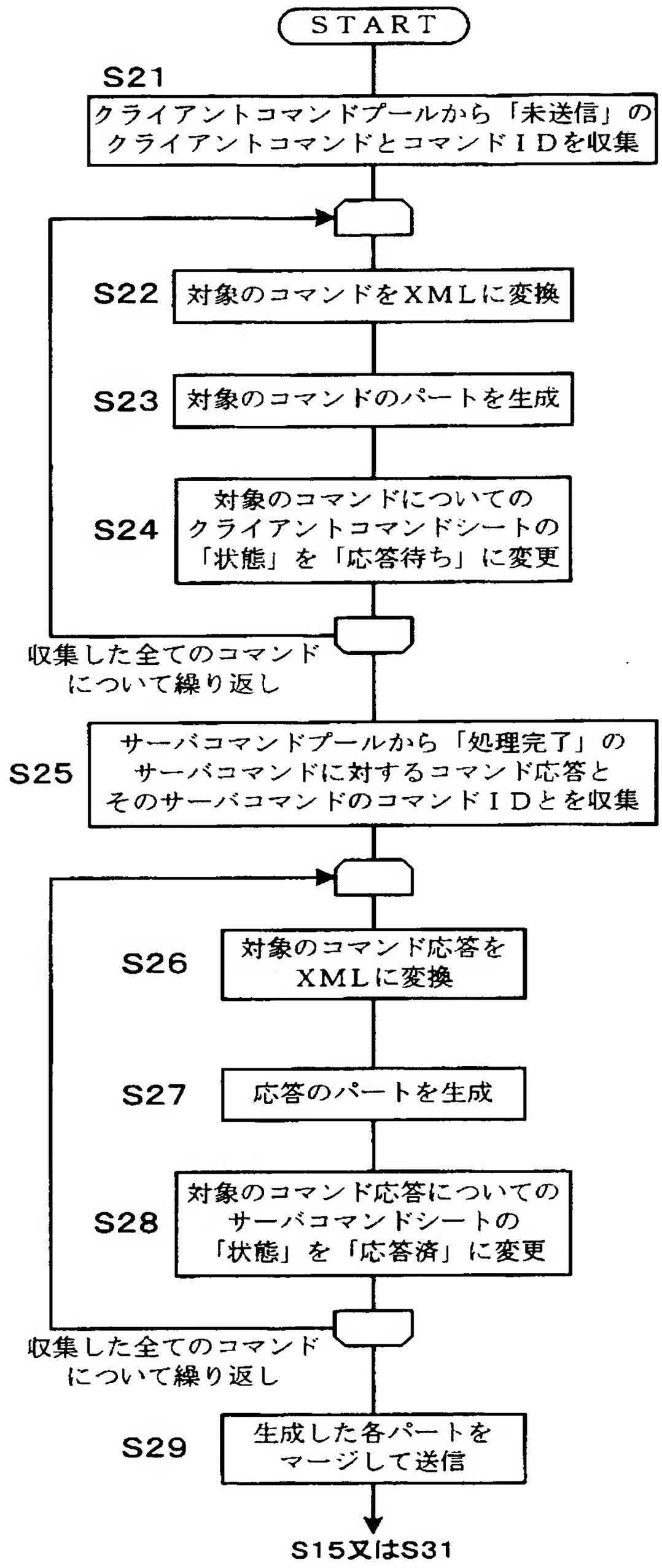
サーバコマンドに対する応答のパート例

```
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: 8bit
X-SOAP-Type: Response

<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:se="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:n="http://www.foo.com/header"
  xmlns:nc:"http://www.foo.com/client"
  s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding">

  <s:Header>
    <n:要求ＩＤ>98765</n:要求ＩＤ>
  </s:Header>
  </s:Body>
    <nc:温度センサ値取得Response>
      <温度>52</温度>
    </nc:温度センサ値取得Response>
  </s:Body>
</s:Envelope>
```

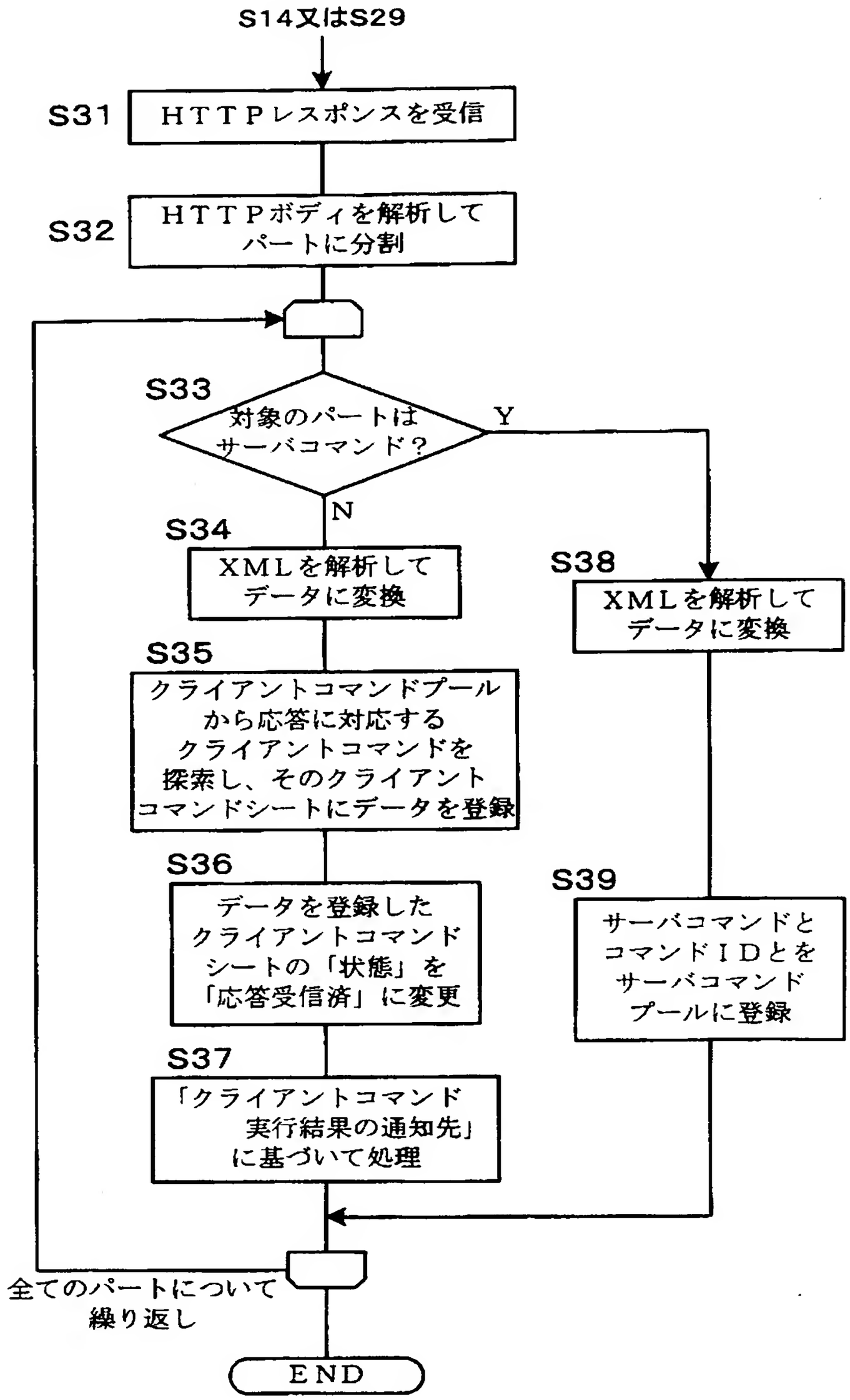
【図１７】


START
S11
クライアントコマンド収集処理
S12
サーバコマンド実行結果
収集処理
S13
各パートを１つにマージ
S14
ＨＴＴＰリクエストを送信
S15
ＨＴＴＰレスポンスを受信
S16
パートに分割
S17
対象のパートは
サーバコマンド？
Y
N
S18
サーバコマンド
登録処理
S19
応答通知処理
全てのパートについて
繰り返し
END

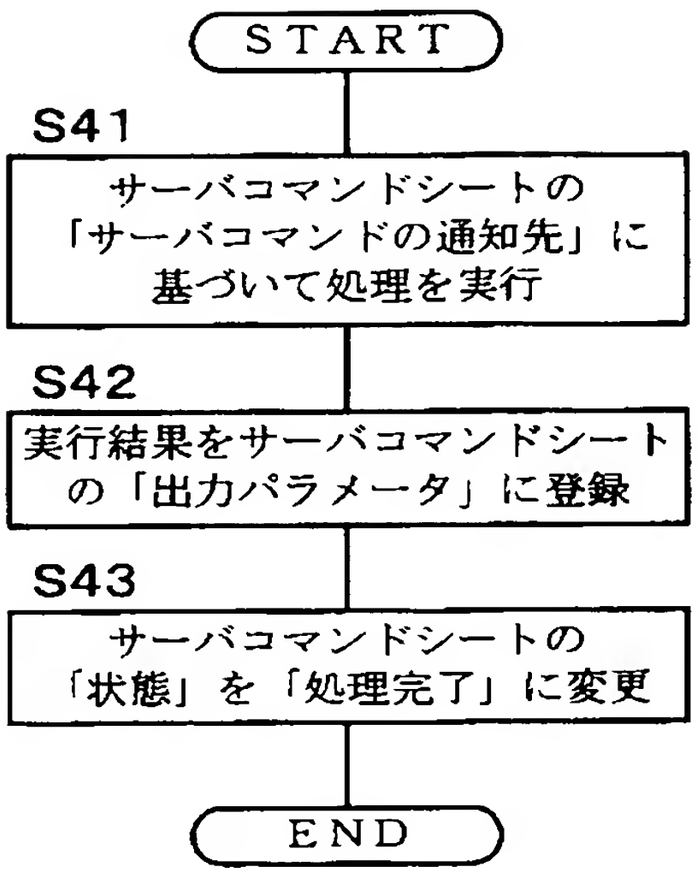
【図１８】


START
S21
クライアントコマンドプールから「未送信」のクライアントコマンドとコマンドＩＤを収集
S22
対象のコマンドをＸＭＬに変換
S23
対象のコマンドのパートを生成
S24
対象のコマンドについてのクライアントコマンドシートの「状態」を「応答待ち」に変更
収集した全てのコマンドについて繰り返し
S25
サーバコマンドプールから「処理完了」のサーバコマンドに対するコマンド応答とそのサーバコマンドのコマンドＩＤとを収集
S26
対象のコマンド応答をＸＭＬに変換
S27
応答のパートを生成
S28
対象のコマンド応答についてのサーバコマンドシートの「状態」を「応答済」に変更
収集した全てのコマンドについて繰り返し
S29
生成した各パートをマージして送信
S15又はS31

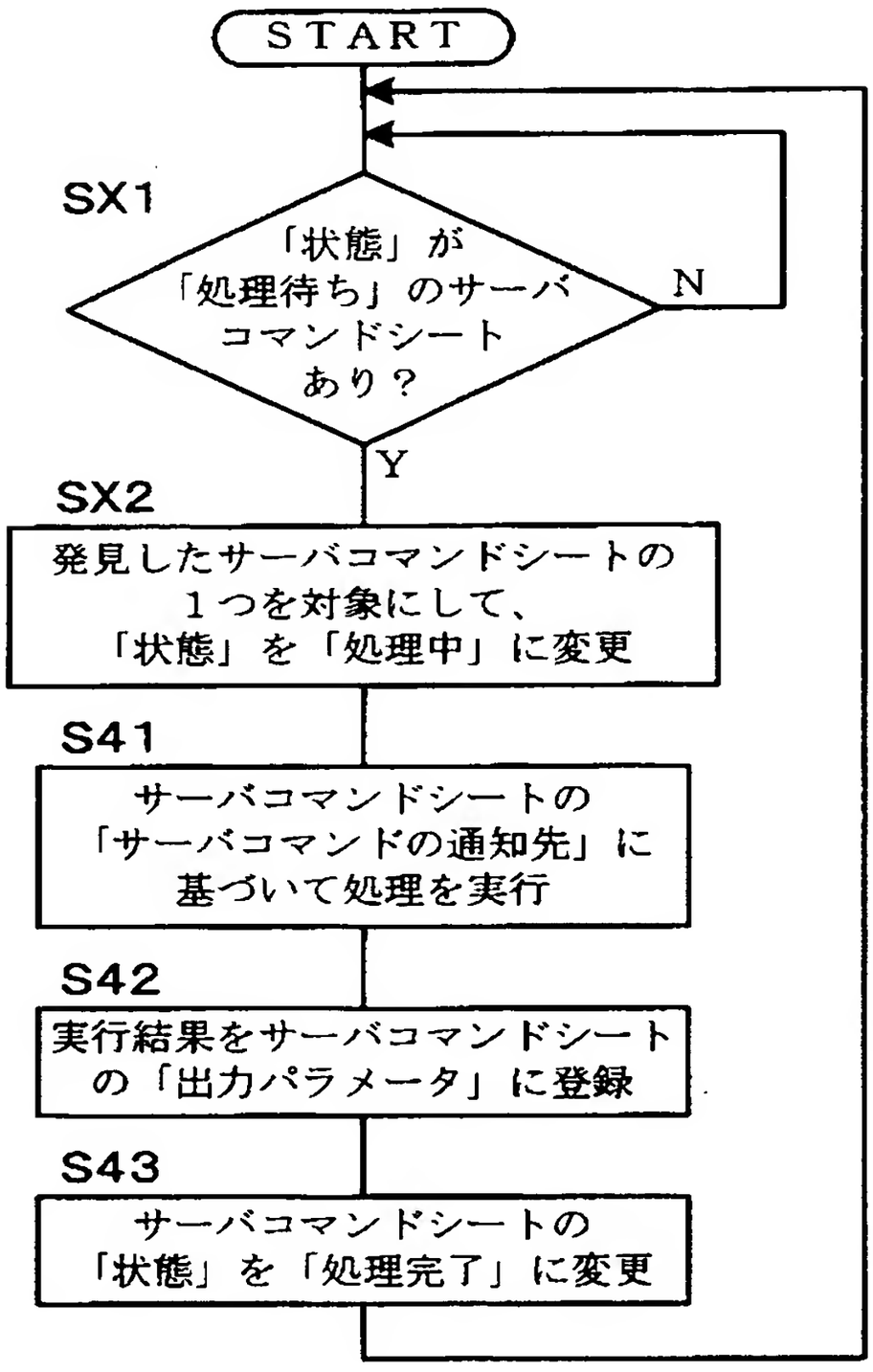
【図１９】


S14又はS29
S31
ＨＴＴＰレスポンスを受信
S32
ＨＴＴＰボディを解析して
パートに分割
S33
対象のパートは
サーバコマンド？
Y
N
S34
ＸＭＬを解析して
データに変換
S35
クライアントコマンドプール
から応答に対応する
クライアントコマンドを
探索し、そのクライアント
コマンドシートにデータを登録
S36
データを登録した
クライアントコマンド
シートの「状態」を
「応答受信済」に変更
S37
「クライアントコマンド
実行結果の通知先」
に基づいて処理
S38
ＸＭＬを解析して
データに変換
S39
サーバコマンドと
コマンドＩＤとを
サーバコマンド
プールに登録
全てのパートについて
繰り返し
END

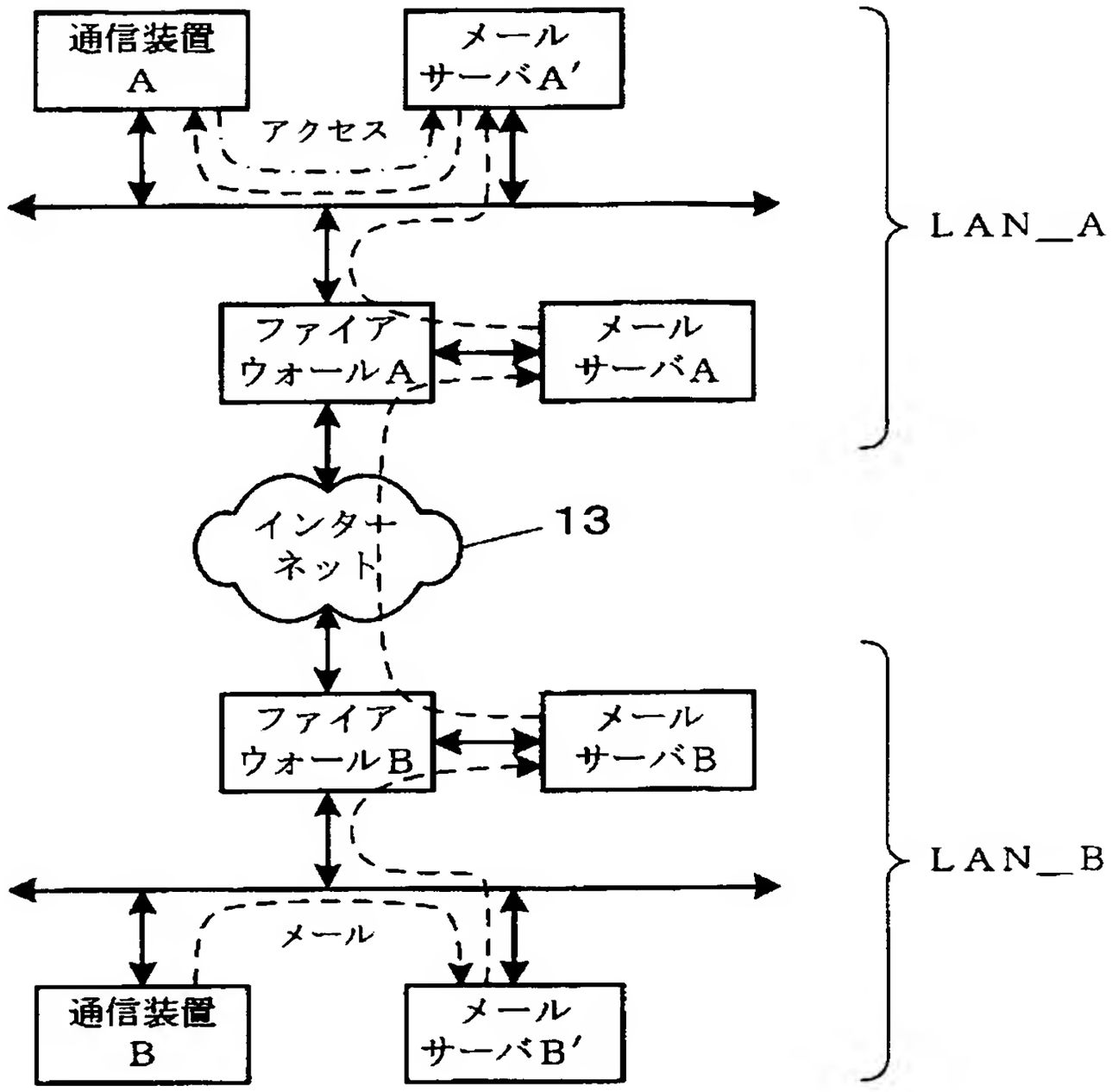
【図２０】


START
S41
サーバコマンドシートの
「サーバコマンドの通知先」に
基づいて処理を実行
S42
実行結果をサーバコマンドシート
の「出力パラメータ」に登録
S43
サーバコマンドシートの
「状態」を「処理完了」に変更
END

【図２１】

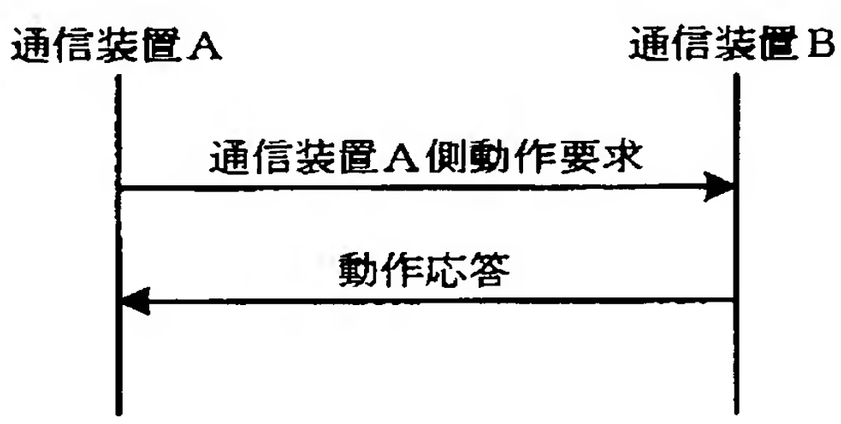

START
SX1
「状態」が
「処理待ち」のサーバ
コマンドシート
あり？
N
Y
SX2
発見したサーバコマンドシートの
１つを対象にして、
「状態」を「処理中」に変更
S41
サーバコマンドシートの
「サーバコマンドの通知先」に
基づいて処理を実行
S42
実行結果をサーバコマンドシート
の「出力パラメータ」に登録
S43
サーバコマンドシートの
「状態」を「処理完了」に変更

【図２２】

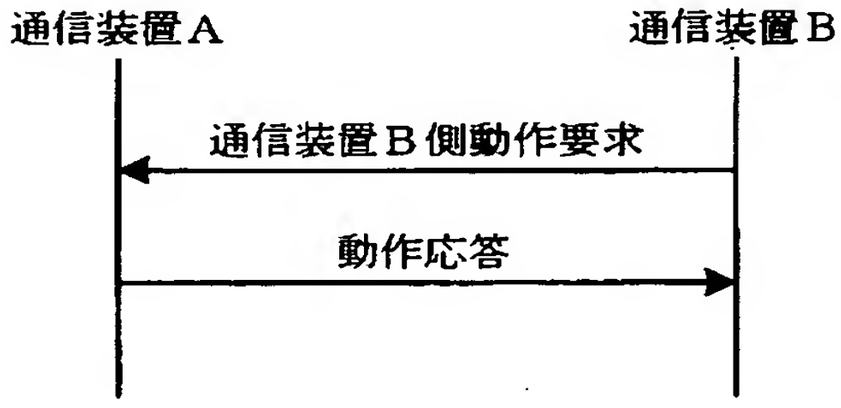

通信装置
A
メール
サーバA′
アクセス
LAN_A
ファイア
ウォールA
メール
サーバA
インタ＋
ネット
13
ファイア
ウォールB
メール
サーバB
LAN_B
メール
通信装置
B
メール
サーバB′

【図２３】

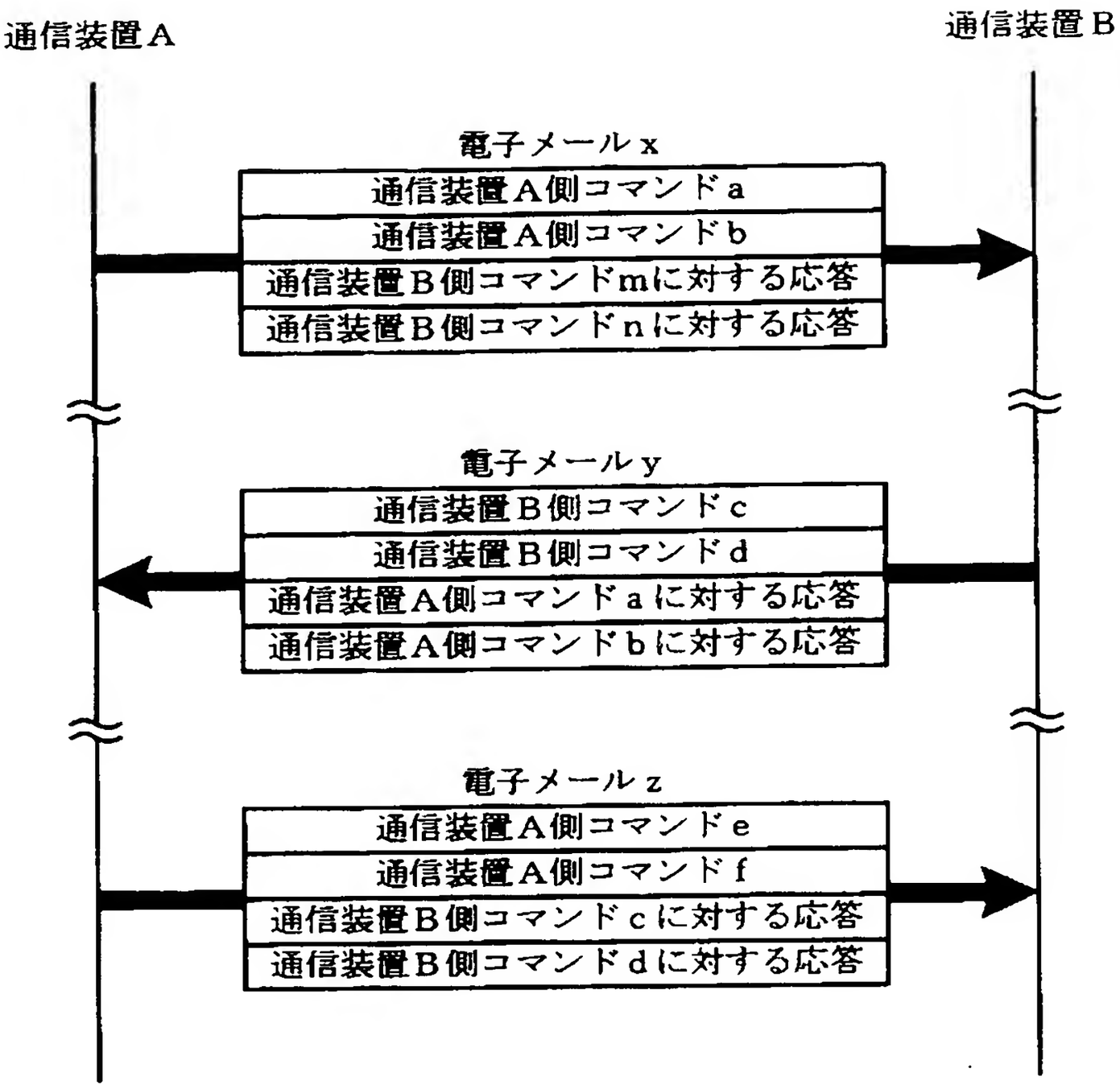
（A）


通信装置A
通信装置B
通信装置A側動作要求
動作応答

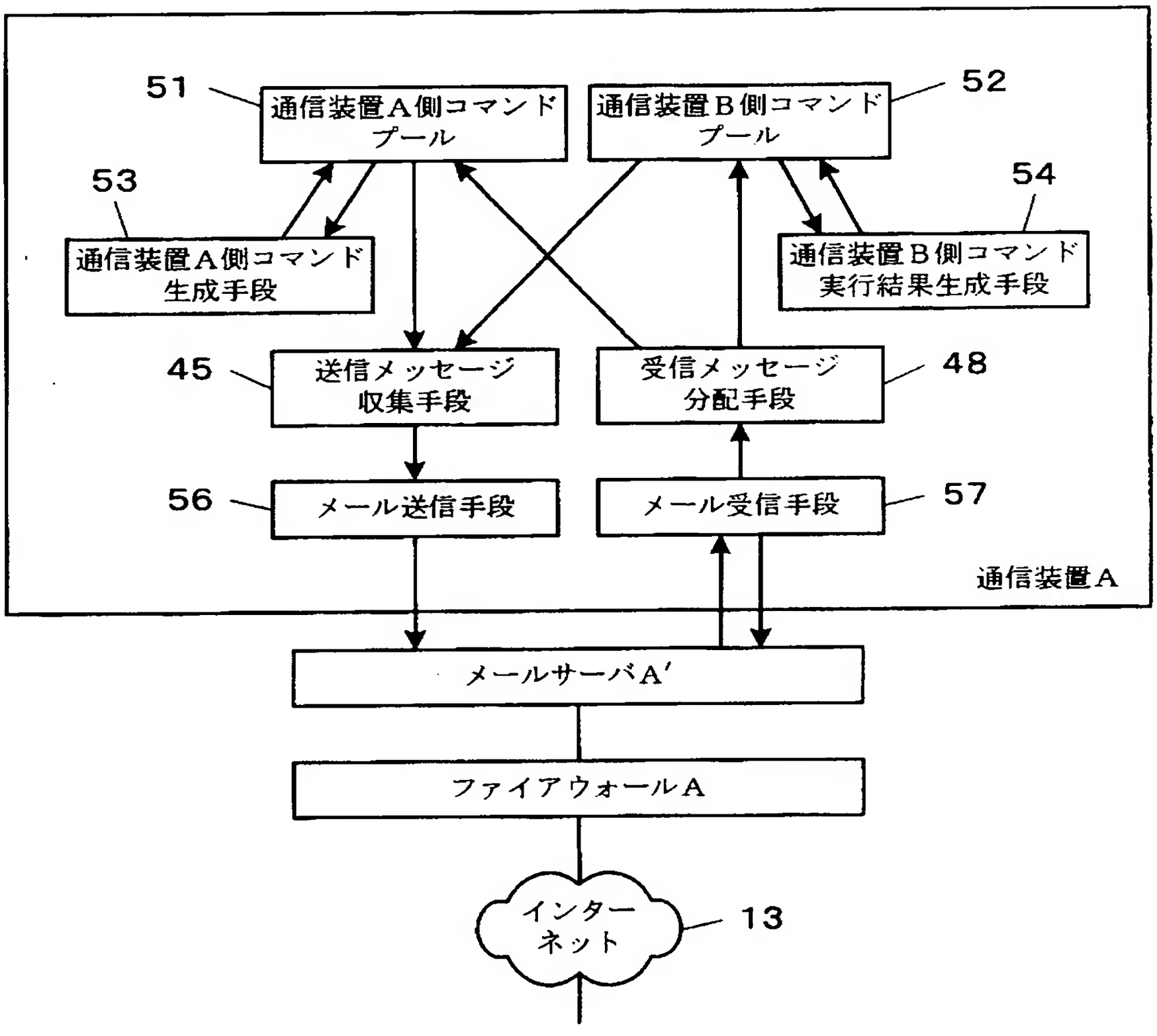
（B）


通信装置A
通信装置B
通信装置B側動作要求
動作応答

【図２４】


通信装置Ａ
通信装置Ｂ
電子メールｘ
通信装置Ａ側コマンドａ
通信装置Ａ側コマンドｂ
通信装置Ｂ側コマンドｍに対する応答
通信装置Ｂ側コマンドｎに対する応答
電子メールｙ
通信装置Ｂ側コマンドｃ
通信装置Ｂ側コマンドｄ
通信装置Ａ側コマンドａに対する応答
通信装置Ａ側コマンドｂに対する応答
電子メールｚ
通信装置Ａ側コマンドｅ
通信装置Ａ側コマンドｆ
通信装置Ｂ側コマンドｃに対する応答
通信装置Ｂ側コマンドｄに対する応答

【図２５】


51
通信装置Ａ側コマンドプール
52
通信装置Ｂ側コマンドプール
53
通信装置Ａ側コマンド生成手段
54
通信装置Ｂ側コマンド実行結果生成手段
45
送信メッセージ収集手段
受信メッセージ分配手段
48
56
メール送信手段
メール受信手段
57
通信装置Ａ
メールサーバＡ′
ファイアウォールＡ
インターネット
13

【図２６】

```
From: deviceA@foo.com
To: deviceB@bar.com
Subject: Multi Message #00000001
Date: Wed, 30 Jul 2003 10:00:00 +0900
Content-Type: multipart/mixed;boundary=MIME_boundary
Content-Length:nnnn

--MIME_boundary
Content-Type:text/xml;charset=iso-2022-jp
Content-Transfer-Encoding:7bit
SOAPAction:"SOAP Action URI"
X-SOAP-Type:Request

<s:Envelope>
  <--SOAP Request-->
</s:Envelope>

--MIME_boundary
Content-Type:text/xml;charset=iso-2022-jp
Content-Transfer-Encoding:7bit
X-SOAP-Type:Response

<s:Envelope>
  <--SOAP Response-->
</s:Envelope>

--MIME_boundary
Content-Type:text/xml;charset=iso-2022-jp
Content-Transfer-Encoding:7bit
X-SOAP-Type:Response

<s:Envelope>
  <--SOAP Response-->
</s:Envelope>

--MIME_boundary
Content-Type:text/xml;charset=iso-2022-jp
Content-Transfer-Encoding:7bit
X-SOAP-Type:Response

<s:Envelope>
  <--SOAP Response-->
</s:Envelope>

--MIME_boundary--
```

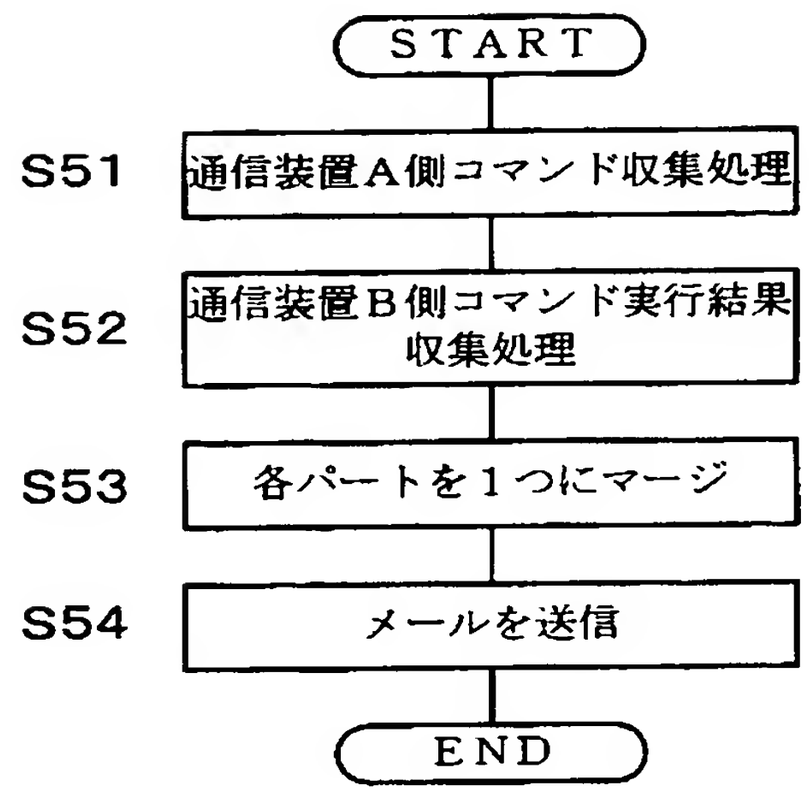
第１パート

第２パート

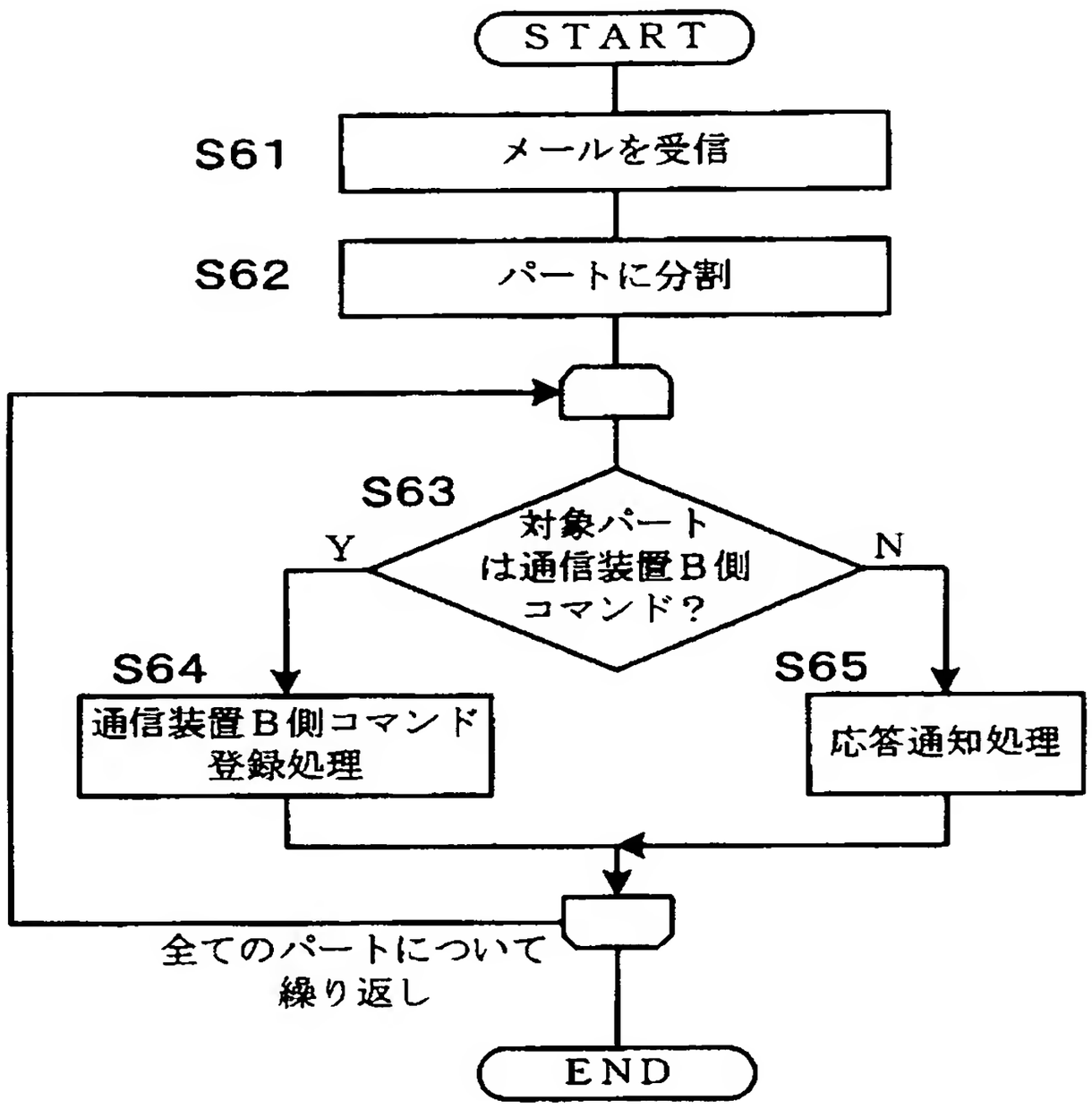
第３パート

第４パート

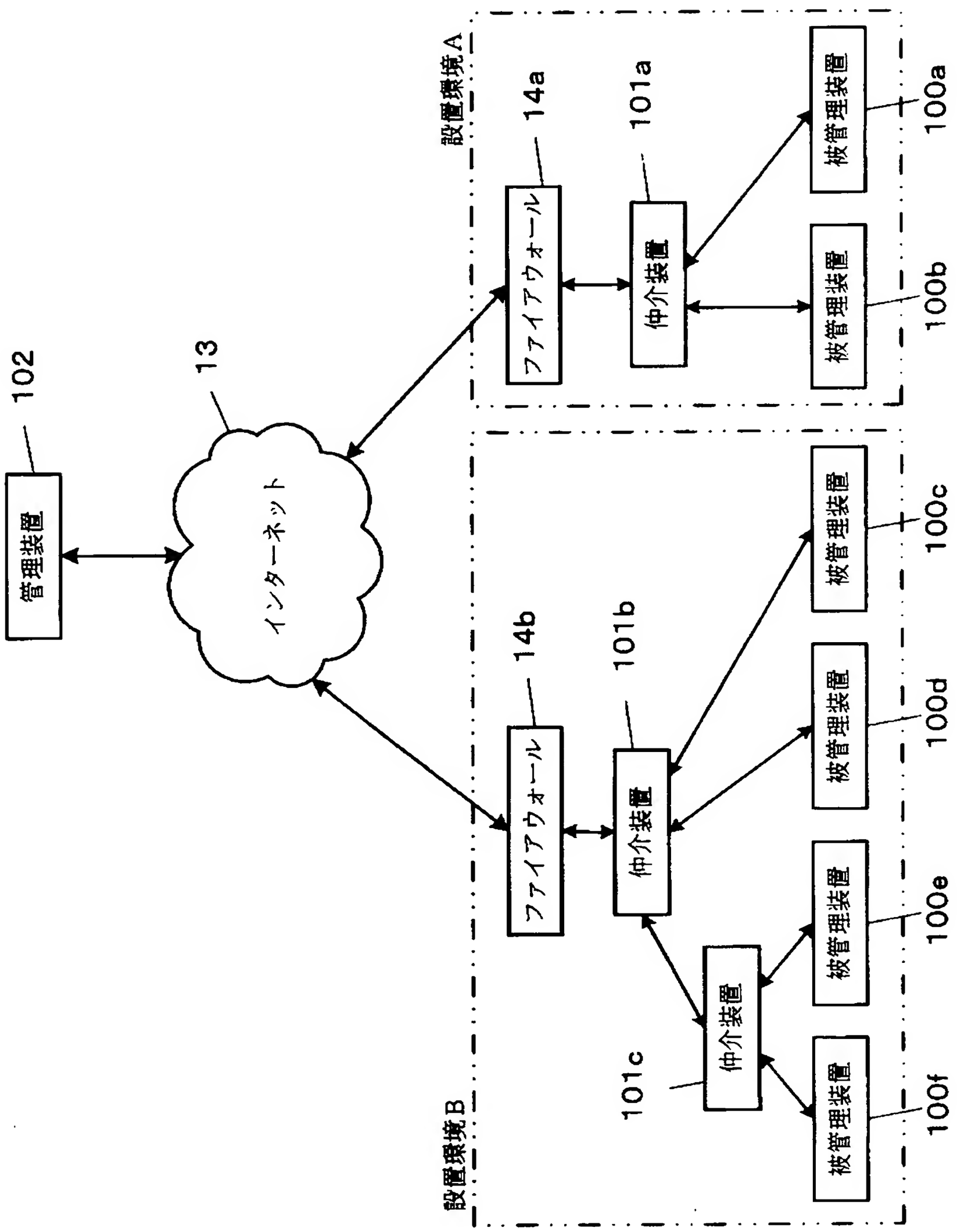
【図２７】


START
S51
通信装置A側コマンド収集処理
S52
通信装置B側コマンド実行結果収集処理
S53
各パートを１つにマージ
S54
メールを送信
END

【図２８】


START
S61
メールを受信
S62
パートに分割
S63
対象パートは通信装置B側コマンド？
Y
N
S64
通信装置B側コマンド登録処理
S65
応答通知処理
全てのパートについて繰り返し
END

【図２９】


管理装置
インターネット
13
102
設置環境A
ファイアウォール
14a
伸介装置
101a
被管理装置
100a
被管理装置
100b
設置環境B
ファイアウォール
14b
伸介装置
101b
被管理装置
100c
被管理装置
100d
伸介装置
101c
被管理装置
100e
被管理装置
100f

【書類名】要約書
【要約】
【課題】　複数の通信装置が互いに動作要求及び受信した動作要求に対する動作応答を送受信する場合において、通信の効率を上げる。
【解決手段】　ＨＴＴＰクライアント１１が、ＨＴＴＰサーバ１２に送信すべきクライアントコマンドとＨＴＴＰサーバ１２から受信したサーバコマンドに対する応答とを一括して１つのＨＴＴＰリクエストとしてＨＴＴＰサーバ１２に送信し、ＨＴＴＰサーバ１２が、ＨＴＴＰクライアントに送信すべきサーバコマンドとＨＴＴＰクライアント１１から受信したクライアントコマンドに対する動作応答とを一括して１つのＨＴＴＰレスポンスとしてＨＴＴＰクライアント１１に送信する。この場合において、各コマンドが関数呼び出しであり、これに対する応答が、その関数呼び出しによって呼び出された関数の実行結果であるとよい。
【選択図】　図５

特願２００３－３０５５１３

出　願　人　履　歴　情　報

識別番号　　［０００００６７４７］

１．変更年月日　　２００２年　５月１７日
［変更理由］　　住所変更
住　所　　東京都大田区中馬込１丁目３番６号
氏　名　　株式会社リコー